Fisheries New Zealand
Tini a Tangaroa

A new Spatially-Disaggregated Agent-Based Simulation Modelling Tool for use in Fisheries Management Strategy Evaluation

New Zealand Fisheries Assessment Report 2018/34

J.R. McKenzie,
N. Bentley,
R. Bian

ISSN 1179-5352 (online)
ISBN 978-1-77665-959-3 (online)

August 2018

New Zealand Government

Requests for further copies should be directed to:

Publications Logistics Officer
Ministry for Primary Industries
PO Box 2526
WELLINGTON 6140

Email: brand@mpi.govt.nz
Telephone: 0800 00 83 33
Facsimile: 04-894 0300

This publication is also available on the Ministry for Primary Industries websites at:
http://www.mpi.govt.nz/news-and-resources/publications
http://fs.fish.govt.nz go to Document library/Research reports

Table of Contents

EXECUTIVE SUMMARY

McKenzie, J.R.; Bentley, N.; Bian, R. (2018). A Spatially-Disaggregated Agent-Based Simulation Modelling Tool for use in Fisheries Management Strategy Evaluation
***New Zealand Fisheries Assessment Report 2018/34.* 48 p.**

Agent-based models have distinct structural advantages over the more "traditional" fisheries partition models for stock assessment simulation, these include: true age-length integration; high spatial complexity; agent memory, i.e. retention of agent "characteristics" across model partition boundaries.
An Agent-Based simulation model was developed primarily for use in model-based fisheries management strategy evaluation (MSE). The AB model core dynamics were designed to meet the simulation requirements of the 2013 SNA 1 assessment model, including three-stock spatial disaggregation and home-fidelity movement. Pursuant to the SNA 1 assessment the AB model is capable of generating CPUE, catch at-age and length, and tag release and recovery observations.

The AB simulation model was coded in C++ and using source-code from the Boost and Stencila repositories for agent memory management and pseudo random number generation.

Currently the AB model can be configured to simulate the exploitation history and productivity dynamics of up to three unit stocks and can accommodate mixing and movement between them. The user defines the model exploitation history in the form of either an exploitation rate or annual catch (TAC), with a current upper limit of 115 harvest years. The model always commences in an equilibrium virgin state at the start of each exploitation run. The model currently supports simultaneous harvest by up to four fishing gears of differing selectivity. The user specifies the productivity dynamics of each unit stock using population-level parameterisations typical of most fisheries assessment models (i.e. growth, natural mortality, steepness, sex, age-at maturity, and virgin biomass); stochasticity is introduced at the agent level. As well as providing the "true" stock status for any given time-step (e.g. numbers at age/length, spawning stock biomass, etc.), the model can generate fishery dependent observational data for any specified time-step in the form of catch at-age, catch at-length and CPUE indices. The model also supports tagging, and is capable of simulating tagging observational data from multiple tag release and recovery events.

The AB model code was tested and validated against similarly parameterised CASAL assessment models and a Petersen tagging estimator model. An identically parameterised length-based CASAL model produced length compositions and stock biomass trajectories that closely matched those of the AB model. Challenged with estimating growth, selectivity and virgin biomass (B0) on the basis of AB model generated length compositional data and CPUE indices, CASAL was able to accurately estimate these parameters and thus again closely predict the AB model stock status. For the final CASAL test, in addition to estimating growth, selectivity and B0 an age-based CASAL model was also tasked with estimating AB model recruitment variation (year-class strength) across 37 consecutive model years. Again CASAL was able to closely estimate the AB model true parameter space (including the 37 recruitment deviates) and thus again was able to achieve an accurate prediction of AB model stock status. To test the AB model tagging and movement dynamics an AB model was configured to generate tag recovery data from four consecutive years after a single tagging event in each of the three stock areas. The AB model was also configured for markovian movement between the three stock areas. The tagging data was provided to a Petersen mark-recapture estimator to enable it to derive population estimates for each of the three stock areas at the time of tagging. In order for the Petersen estimator to correctly estimate AB model population size it would first need to "correct" the tag recovery data for growth and movement thus there was a prerequisite requirement for the Petersen estimator to estimate these parameters. The Petersen population estimates derived from growth and movement corrected tagging data closely matched the AB model true population numbers, the inference being that the AB model tagging and movement dynamics are sound.

The current AB model is suitable as an operating model for SNA 1 MSE, for generating tagging data to evaluate SNA 1 mark-recapture designs, and for testing SNA 1 stock assessment model performance and bias. With only minor structural changes the current AB model simulator could also be used to undertake model-based MSEs for most of New Zealand commercial fish stocks including hoki. In future development of the AB model, it would be desirable to add fine-scale spatial partition capability in addition to the current stock-scale partition structure. Doing this would allow more complex spatial management scenarios to be simulated (e.g. closed areas, spatial gear restrictions and marine protected areas).

1 WHAT ARE AB MODELS AND HOW ARE THEY USEFUL FOR FISHERIES SIMULATIONS?

This document provides a technical description of an Agent-Based Model (AB model) software tool developed by NIWA and Trophia intended to build fish stock simulation models for evaluating the performance of age or length-based stock assessment models and for building "real-world" construct operating models for use in Management Strategy Evaluation (MSE).

The AB modelling tool described in this document was built to mimic the stock assessment requirements of the two main New Zealand snapper (*Pagrus auratus*) stocks (SNA 1 and SNA 8; MPI 2017). The current AB model is capable of integrating the complex spatial and mixing dynamics evident in these stocks, and thus simulating observational data required by the current SNA 1 and SNA 8 assessment models, specifically: mark-recapture (tagging); commercial catch-at-age and length, CPUE indices, and recreational harvest estimates. Although the AB model is capable of mimicking complex stock assessment model configurations, simple stock model parameterisations are also supported. We believe the base AB model software has a wide range of management applications beyond MSE, for example: the exploration of optimum equilibrium yield under different gear selectivity, discard survival and minimum legal size scenarios; evaluating stock assessment model estimation performance. The base model c++ code resides in the public domain and can be downloaded from the following repository: https://github.com/trophia/sna1.

Agent-based and Individual-based models (ABMs & IBMs) follow the fate of individuals through their life cycle, under the assumption that individual behaviour has an appreciable effect on population dynamics (Grimm & Railsback 2005). IBMs and ABMs are largely functionally synonymous, the key difference being that IBMs model true individuals whereas ABMs model "agents" each agent representing a group of individuals having the same life history characteristics and fates. AB models can accommodate more complex formal partition space than more "traditional" partition models (effectively, the number of partitions in an AB model is only limited by the number of agents). Partitions such as length, age, spatial location, growth dynamics and tag status, are all defined as attributes of either an individual fish or a group of fish that share the same attributes and fates (e.g. location age length, sex, growth-rate, tagging status), i.e. "agents". The system level properties emerge from these lifecycle traits and the behaviour of the individuals. For example, the frequency of individual agents in the AB model for any point and any desired partition structure is simply derived by summing across fish attribute classes, e.g. length frequency of fish in area 'A' that is tagged. At each time-step, the fate of each fish is governed by defined probabilities in accordance to their inherent/unique set of attributes (e.g. move, grow, or caught by a fishery). When AB model agents are defined to represent groups of fish rather than an individual, weight-scalars can be used to transform model outputs and observations to reflect "real world" biomass, catches, tag recoveries, and harvest targets.

The second powerful feature of AB models that distinguishes them from partition models is "Agent memory", meaning that agent attributes (e.g. length, age, growth-rates, etc) are retained with the agent as it moves between model spatial-temporal partitions. The agent memory aspect of AB models enables them to simulate "true" age/length dynamic integration (c.f. standard assessment models which tend to be either age or length based but seldom both).

1.1 AB model advantages, disadvantages and core purpose

AB models include aspects usually ignored in analytical models, such as variability among individuals, complete life cycles, and spatial movements, which make them necessarily more complex in structure. This increased complexity is largely achieved through increasing the number of agents, thus increasing computation time.

AB model Advantages:

- True length and age-based model
- Retains attributes of a particular cohort of fish through time
- Allows incorporation of greater partition space than other modelling approaches
- Efficient at modelling complex spatial complexity and movement

AB model Disadvantages:

- Number of individuals in the model computationally limiting
- May need to use a scaler to model large populations, individual agents being representative of a group of fish that share the same fates and attributes
- Computational speed limitations mean AB models are typically not viable for estimation.

AB models core purpose:

- As an operating model for assessing the performance of more “traditional” age or length based assessment models (estimators) and as a “real-world” construct in Management Strategy Evaluation simulation modelling.

2 AB MODELLING TOOL SPECIFICATIONS

2.1 AB Model Overview

2.1.1 Dynamics overview

The AB model stock is universally populated by individual fish (i.e. agents) that are born, grow, move, are tagged and die across the model’s spatial-temporal partition space. The status of the stock and its spatial distribution can be derived at any time-step by summing the agents and the agent attributes.

A recruitment process governs the number of fish recruiting to specific spatio-temporal partitions. The individual recruits coming from a defined length-frequency distribution, all being the same initial recruited age (e.g. 1-year-old, 3 months post-settlement etc). AB model growth is implemented as a set of agent-specific length increments in each model time-step in accordance to either a linear or exponential user-parameterised growth increment model. Individual movement across the spatial-temporal partition space is governed by predefined movement matrices, the underlying dynamics being either Markovian or Home Fidelity (Appendix 1).

Population-level and agent-level movement processes are implemented in a similar way to growth, i.e. individual agents being assigned their own unique set of movement probability parameters at “birth”, these drawn from population-level movement parameter prior distributions.

There are three main human activities included in the AB model: tag releasing (or tagging), harvest, and tag recovery. In the tagging process, tags are released in specified spatial-temporal partitions in accordance with given selectivity and initial survival rules. The harvest process is pursuant to user-defined quota allocations to up to four fishing methods, subject to minimum legal size (MLS) and selectivity rules. Based on these rules, selected agents are removed from the population in each spatial-temporal partition. Tags are also recovered during harvesting subject to tag detection and scanning rules.

The AB model dynamics are governed by “higher-level” population processes parameterised in a similar way to most "typical" stock assessment population models, i.e. mean recruitment, natural mortality, mean growth, total harvest rules, gear selectivity ogives, discard mortality rates, etc. The AB model population-level recruitment processes can be configured to conform to either the Beverton and Holt or [1]Ricker stock-recruit dynamic.

2.1.2 The role of “chance”

Almost all population level dynamics in AB models are implemented as sequences of chance outcome tests (“coin tosses”) enacted at the individual agent level (i.e. stochasticity) in accordance with a series of functionally-determined probabilities. For example, natural mortality is usually implemented as an annual rate in standard partition models. The AB model accounts for the stochastic nature of the natural mortality process, and its application enacted via successive “coin tosses” across all model agents currently “alive” in the given time-step in accordance with the probability of the given agent dying of natural causes in that time-step. Probability “heads” the agent lives, at least until the next time-step; probability “tails” it dies and leaves the model. Although it is impossible to know *a priori* which agents will die of natural causes, the laws of probability will ensure that the AB model replicates the “desired” proportional loss of agents due to natural mortality at the population-level.

Due to the fundamental role chance plays in AB model dynamics, AB model performance is strongly determined by the degree of inherent bias in the pseudorandom number generation processes (PRNG). The current AB modelling tool uses PRNG algorithms sourced from the gcc Boost Library (http://boost.sourceforge.net/libs/random/index.html) - these algorithms are often used for randomisation applications because their auto-correlation, sequence repetition and bias generation levels are extremely low (Matsumoto & Nishimura 1998).

2.2 AB model spatial-temporal structure and population dynamics process order

2.2.1 Spatial-temporal partition structure

Stock-management-area-partition

This defines the stock spatial management unit of interest being the one for which quotas are set, stock assessments undertaken and management decisions made. The current AB model can be configured with up to three stock management areas/regions and biological stocks (note: the intention is to increase the maximum number of stock management partitions in future AB model developments). In the AB model, there is always the same number of stocks as the stock-management area partitions. Where there is either zero or Markovian (Appendix 1) movement between areas, AB model management areas and AB model stocks are always synonymous. However, under home-fidelity movement (Appendix 1) it is possible in the AB model for stocks and management areas to be decoupled, i.e. there is more than one biological stock present in a given management area.

[1] Currently only Beverton and Holt implemented

Sub-region-partition

This is the base spatial resolution unit in the model - the stock-management-area-partition potentially being made up of one or more sub-regional cells. The AB model can only currently accommodate one sub-regional-partition per stock management area, i.e. sub-regional-partitions and stock management partitions are effectively synonymous in the current model structure. The concept of a sub-regional partition is introduced here because the intention is to add multiple sub-regional partition capability to the AB model in the future (Section 5.5).

Annual cycle

The AB model time cycle "of interest" to management is typically annual, which is of the same temporal scale as a stock assessment. The AB model reports cumulative statistics relative to the end of the annual cycle.

Time-step

The annual cycle can be divided up into a number of user-defined discrete time-steps (e.g. season, month day, etc). In the AB model the fates of all fish in the model are updated in each time-step, therefore the number of time-steps in the annual-cycle has a direct bearing on model processing speed (or time to complete a model run). The AB model is currently "hard-wired" with one annual time-step. It is envisaged that future AB model developments will enable the user to sub-divide the annual cycle at least down to a monthly time interval level (Section 5.2).

2.2.2 Within time-step model process order

The key model processes occurring with each time-step are as follows:

Individual agent update process:

- Update individual agent attributes (requires the model to step through every live agent)

Tagging process **(process occurs during a designated tag release time-step)**

- Accumulate current model population statistics (e.g. numbers at length/age by area etc) and derive length-based tagging probabilities by gear and region
- Tagging process: random draw to the required number of tagged fish from the population.

Harvest and tag recovery process

- Accumulate current model population statistics (e.g. numbers at length/age by area etc) and derive length-base harvest probabilities by gear and region

- Harvest process: random draw to the required number of harvested fish from the population and recover tags.

Model output process

- Model output process: Accumulate end of time-step model population statistics (e.g. numbers at length/age by area etc.) and generate required output statistics including observational data for input to an estimator model.

2.2.3 Within time-step individual fish process order

1. Recruitment (at initial age = 0)
2. Ageing (initial age + timestep increment time)
3. Maturation (currently implemented as a size-based probability ogive)
4. Stock-region Migration/movement (Markovian or HF movement rules)
5. Sub-regional movement (currently not implemented)
6. Growth (amount of growth occurring, i.e. growth increment, over time-step *x*)
7. Natural mortality (may be specific to the Sub-area-spatial-partition the fish now resides in)
8. Anthropogenic mortality other than fishing (may be specific to the Sub-area-spatial-partition the fish now resides in)
9. Tag release events (specific to the Sub-area-spatial-partition the fish now resides in)
    - release mortality (specific to capture method)
    - reset growth rate (specific to capture method)
    - tag loss rate (specific to tag type)
10. Harvest induced mortality and tag recovery (specific to the Sub-area-spatial-partition the fish now resides in)

2.2.4 AB Model run phases

1. Burn in period: necessary to achieve virgin equilibrium conditions (currently set at 200 annual cycles).

2. Catch history period: A catch history is applied to achieve the required MSE/projection starting point conditions.

3. MSE/projection period: time-steps over which MSE simulations take place.

Note: For simulations where the goal is simply to investigate estimation precision and bias in stock assessment models, stock projections (step 3) may not be required.

2.3 AB Model structural specifications currently implemented (and proposed)

As with most fisheries models, AB Models are fundamentally specified by population-level parameterisations (e.g. mean growth, mean recruitment, natural and fishing mortality). Although most AB model dynamics are enacted stochastically at the individual agent level, for the AB model to "work" as intended, agent-level stochasticity must be consistent with population-level parameterisations and priors.

2.3.1 AB model definition of a biological stock

In the AB model, stocks are defined in reference to stock-management-area-partition definitions. In most stock assessment applications stocks and areas are synonymous. In the AB model stock-area synonymy holds even when there is mixing between stock areas movements as long as the type of movement is diffusive or Markovian (Appendix 1). The reason being that under diffusive movement, once a fish agent moves into an area, the chance of it leaving at a future time is identical to the leaving probability of a fish agent that was spawned in the area and still currently resides there. When the AB model is configured for Markovian movement the biomass of a given stock is simply derived by summing up the weights of all agents currently residing in a given stock area regardless of into which stock area the agents were originally recruited.

Stock and stock-area synonymy breaks down under home-fidelity movement (Appendix 1). Under home-fidelity movement the probability of a fish agent leaving a given stock area can vary depending upon which stock area it originally recruited to. When configured for home-fidelity movement the AB model definition of a biological stock becomes “a population of agents sharing the same natal stock recruitment area”. Under home-fidelity movement AB model stock biomasses are derived by summing across fish agents relative to their stock recruitment areas. The AB model can be requested to output population summary statistics relative to stock spatial areas instead of biological stock. Note: AB model observational/monitoring data (see below), regardless of movement dynamic specification, are only generated specific to stock spatial areas and/or regional partitions.

2.3.2 Virgin spawning stock biomass (B0) tonnes

The virgin spawning stock biomass (B0), otherwise known as the maximum stock carrying capacity, is defined by the user for each stock. The AB model B0s are defined specifically to biological stock not by spatial area partition. The AB model currently supports three separate biological stocks requiring three separate B0s to be specified.

2.3.3 Virgin recruitment (R0) numbers

The AB model definition of virgin recruitment (R0) is the number of 0-year-old agents entering the model in time-step *x* of annual cycle *y* under equilibrium virgin (i.e. absence of fishing) conditions. The AB model is able to calculate the number of recruits needed to maintain stocks at the B0 specified levels on the basis of B0 and other user-specified productivity parameters (e.g. growth, natural mortality, and the stock-recruit steepness parameter). As with B0, R0 is specific to biological stock.

Note: Although all new agents enter the model at age 0, after the new recruits are added they, and all other fish in the model, are incremented in age by adding the time-step’s time increment value to their current age; i.e. for new recruits their initial recruitment age is effectively: 0 + time-step increment.

2.3.4 Natural mortality (M)

Currently implemented as a single instantaneous rate (converted to a mortality probability at the time-step time-increment level). A likely future AB model development will be the ability to make M area and age dependent.

2.3.5 Stock recruit (steepness)

The AB model uses a Beverton & Holt (1957) stock-recruitment function to moderate annual recruitment relative to stock biomass; the user can alter the level of stock-recruit dependence by means of the BH model steepness parameter (h). It is intended to include other stock-recruitment functions in future AB model developments, including the Ricker (1954) SR model and dispensatory models.

2.3.6 Sex and maturity

The user can specify the male-female ratio of the number of recruits entering the model. However, in its current form the AB model does not support sex stratification for dynamics such as growth and maturity. Spawning stock biomass (SSB) in the model is currently defined as the weight of all mature agents irrespective of sex. The user is able to define the cumulative sex ratio of recruiting agents entering the model (default 50:50) but at the present the model does not support agents changing sex.

Maturity for both sexes is specified by a user-defined age ogive, which determines the probability of an immature agent becoming mature in a given time-step.

2.3.7 Growth

The amount each agent will grow in a given time-step is determined by a user-parameterised growth function. The AB model currently supports two growth model specifications:

1. Linear increment model (von Bertalanffy);
2. Exponential increment model (Appendix 1).

The AB model can be configured to achieve growth stochasticity at the population level in three ways:

1. Agent growth variability: Individual agent growth follows a deterministic trajectory, each agent assigned its own unique set of growth parameters at “birth”. Because the individual agent growth parameters are randomly drawn from population-level growth parameter prior distributions the AB model is able to achieve comparable “real world” length at-age means and variability at the population level, rather than purely deterministic based on the age of the individual.

2. Temporal growth variability: All stock agents grow in accordance with the mean population level growth model parameterisation with stochasticity being introduced in the actual increment each agent grows in each time-step

3. As a combination of 1 and 2.

The AB model currently allows the user to specify different growth parameterisations and priors for each of the three stocks independently. The model agents can also be configured to adopt a new random growth path after moving to a stock area having a different mean growth rate from the area they came from.

Other parameterisation we see is important to add in the near future are:

- Sex partitioned growth
- Changing agent growth rates as a consequence of being tagged
- Density-dependent growth

Note: Similar to ageing, new recruits enter the model at length 0, after which they, and all the other model agents, are incremented in length by adding the time-step's growth increment value to their current length; i.e. for new recruits their initial recruitment length is effectively: 0 + time-step length increment. Because population statistics are derived after ageing and growth have occurred (i.e. after step 7 in Section 2.2.3) there are never any zero length (or age) agents in model summary statistics.

2.3.8 Anthropogenic mortality

The user has the option to specify a second mortality rate to account for anthropogenic sources of mortality other than fishing. Anthropogenic mortality is currently specified and internally implemented identically to natural mortality.

2.3.9 Movement

The rules and processes that govern movement are set at the population level and both home fidelity and Markovian (Appendix 1) dynamics are possible. Movement is specified using *n* x *n* movement probability matrices where *n* is the number of stock area partitions in the model (currently 3). The user specifies movement as the probability of movement over the course of an annual cycle. Note: adding fraction-year time-steps in a future iteration of the AB model will require the AB model to derive movement probability matrices specific to the year-fraction time interval.

2.3.10 Gear selectivity

Gear selectivity is defined at the population level as functions (ogives) or matrices (note: currently only double normal selectivity parametrisation is accommodated). All selectivities used in the AB model are length-based. The effect of selectivity is implemented at the individual level as being the probability of an agent of given length being captured by specific gear in a specific region in a specific time-step. In the AB model, selectivity serves to modify agent-specific harvest, tagging and discard-mortality probabilities.

2.3.11 Catch removals (harvest)

Harvest (fishing mortality) can be specified either as method-area specific catch vectors (these currently corresponding to annual time-steps) or as method-area specific exploitation rates (F). Effectively the removal or harvest "rules" govern how a given management strategy is implemented in the model. Harvest is implemented at the agent level as the probability of a given agent at a given time being in a given method-area catch mediated by a selectivity ogive. If an agent is caught and if its length is greater than or equal to the user-defined minimum legal size (MLS) it will be designated as "harvested", and the agent will be removed from the model. As harvest probability removals are applied sequentially across all available partition agents, there is potential for bias to be introduced when modelling instantaneous removals by multiple methods of differing selectivities. The AB model is able to replicate instantaneous removals across multiple gear-types by randomising the gear order in which the probability of capture is assessed for each agent (Appendix 3).

2.3.12 Harvest discard (incidental) mortality

A provision for discard mortality is currently implemented as a user-defined single probability parameter being applied to all agents caught by the gear but not harvested (i.e. sub-MLS agents).

2.3.13 Tag release

Tagging is functionally implemented in the AB model in a similar way to harvest but run as a separate process so as to allow the tag release design to differ from the harvest strategy. Instead of harvest, a prescribed number of tags are allocated to a specific gear (or gears) in each region in a given time-step. Tagging at the individual level is governed by the probability of a given fish in a given region at a given time being tagged by a given method which has been allocated a given number of tags.

In the AB model, tagging always occurs prior to harvest making it possible (if the user desires) to recover tags in the same time-step as the release time-step. If this happens, initial tag recoveries are not subject to growth or movement due to these processes occurring before tagging (refer Section 2.2.3).

2.3.14 Tag recovery normal

Tag recovery is also implemented in the AB model harvest cycle, in accordance to tag detection rules and the amount of catch scanned; this is effected as a single probability derived as follows:

P[tag recovered] = P[tag in catch]×P[catch scanned] ×P[tag detected]

2.3.15 Tag recovery trap avoidance

The user can include a method-specific trap avoidance effect in the AB model that will act to reduce the probability of a tagged agent being caught by a given method given that method was also used to tag the agent.

2.4 Individual Agent attributes

Model agents are defined by their unique attributes (i.e. individual traits) some of which are updated in each time-step. In the current AB model all agents have the following stored attributes:

- current age
- maturity (age dependent)
- sex
- current stock-management-area-partition
- recruitment (birth) stock-management-area-partition
- individual growth path parameters
- current length
- time-step, region and method of harvest (only recorded if fish is harvested)
- mortality status (dead or alive)

Tagged fish attributes (discarded if tag is lost)

- tag-number
- tag release time-step
- tag release region
- length at time of tagging
- tag release method

2.5 Population scaling

There are two options for increasing the effective number of fish in the model, so as to achieve equivalence to a typical stock, e.g. scaling 100 000 agents to represent a population of 50 000 000 fish. These are:

1. Constant ratio scaling: all agents represent the same number of "true" individuals;
2. Agent-based scaling: the number of "true" individuals each agent represents can vary.

2.5.1 Constant ratio scaling (currently implemented)

The AB model gets around the problem of needing to create more than 100 000 000 agents to represent a typical fish stock similar to SNA 1 by the use of a single scaling parameter. As a trade-off between computational limits and model realism, the scaling parameter represents the number of true population individuals each agent represents, and this ratio is the same for all agents. The agent scaling parameter is derived by dividing the user-defined equilibrium biomass weight (B0) by the sum of the individual weights of all mature agents in the population at equilibrium (B0) (note: the agent scalar does not have to be an integer).

Having fewer than the "true" number of fish in the AB model reduces its precision, particularly in tagging scenarios where model tags instead represent units of *x* tagged fish not individual tags. The overall effect of scaling is that stochasticity (variability) across the model partition space may be overstated. This is unlikely to be an issue with less complex models but will increase as complexity increases i.e. as the number of model individuals per model partition reduces.

2.5.2 Agent-based scaling (proposed for tagged agents)

We believe the strongest justification in using an agent-based approach over basic scaling is when tagging needs to be a feature of the model that reflects population dynamics (see also Section 5.3). In real tagging scenarios the number of tags typically recovered from the population, relative to its population size, is low, i.e. on the order of hundreds or thousands of tags relative to millions or tens of millions of fish in the population. Constant-ratio scaling applied at the tag recovery level may significantly limit the "effective" resolution power of the observed number of model tag recoveries, e.g. given a scale factor of 50 a recovery of 500 implied tags will have an effective estimation power equivalent to that of only 10 true individual tags. The presence of fine-scale partitioning in the model where the recovery expectation is only one or two tags per partition will exacerbate the problem as scaling is likely to amplify the inherit bias associated with low tag recovery numbers (Seber 1982).

3 AB MODEL CONFIGURATION AND EXECUTION

The base AB model c++ code resides in the public domain and can be downloaded from the following repository: https://github.com/trophia/sna1. Instructions for compiling and running the model code are also available from this site. The model is configured by a series of user input files all of which reside in the sub-directory called ‘input’. Model outputs take the form of a series of appropriately named text files that can found in the sub-directory called ‘output’. An overview of the AB model input and output files is provided below, more detailed descriptions can be found in Appendix 4. All parameters have default values written into the code, and these input files provide a way of optionally overriding these values. All of these parameters are then recorded into the corresponding file in the output folder making it possible to check the values read in and actually used.

3.1 AB model input parameter files (Appendix 4)

`parameters.json:` A JSON file containing single-valued parameters.

`fishes_b0.tsv:` Stock virgin biomass in tonnes.

`fishes_movement.tsv:` Agent {within time-step} stock movement probabilities

`fishes_rec_strengths.tsv` Stock recruitment year class strengths (assumed to be 1 if not specified in this file).

`fishes_shyness.tsv:` Controls the degree of shyness of a fish to the last fishing method that it was caught by (assuming that it was subsequently released because it was undersized or tagged).

`harvest_catch_history.tsv:` Specifies model stock or region annual catch history by method.

`monitoring_programme.tsv:` Specifies what type of monitoring data (observational) is to be output in each given year for all methods and regions. The data output types being: length frequency (L); age frequency (A) and CPUE (C).

`tagging_releases.tsv:` Specifies the number of tags to release by year, region and method.

`tagging_scanning.tsv:` Specifies the proportion of the annual catch to be scanned by year, region and method.

Currently “gear selectivity”, “age at maturity”, and gear-specific minimum legal size (MLS) can only be altered by editing the “parameters.hpp” c++ file where all the AB model fixed parameters are specified (Appendix 5), and recompiling the program. This will be rectified in the next iteration of the code (see Section 5).

3.2 AB Model output files

The AB model generates two types of output data:

1. AB model true state data: True population spawning stock biomass trajectories; true length and age compositions in each time-step;
2. Stock assessment monitoring data: CPUE indices, catch at-age, catch-at-length, tag recovery observations.

After running the AB model, monitoring, summary statistics and tag recovery information can be found in various text files and sub-directories in the sub-directory “output”.

4 AB MODEL TESTS AND VALIDATION RESULTS

4.1 Testing approach overview

The integrity of the AB model tool was validated against standard stock assessment models constructed in CASAL (Bull et al. 2012) of known performance and complexity, and the premise is that an identically configured AB model and alternate stock assessment model should produce very similar biomass trajectories, age and length distributions. Similarly, an identically structured CASAL estimation model should be able to "predict" the AB model current stock status and biomass history with a high degree of accuracy, when fitted to AB model, simulated 'observational data'. The following comparative tests of the AB model with CASAL standard stock assessment models were made:

1. Three-stock length-based CASAL model identical parameterisation (no fitting)
2. Three-stock length-based CASAL model fitted to AB model generated observational data (i.e. method-region specific length-frequency and CPUE data).
3. Three-stock age-based CASAL model fitted to AB model generated observational data with annual recruitment variation (i.e. method-region specific age-frequency and CPUE data).
4. Petersen tag estimation model predictions of recruited stock size (numbers) growth and movement from AB model generated single tagging event observational data.

4.1.1 Model Specifications

An AB model with similar spatial structure, productivity dynamics and catch history to the 2015 SNA 1 assessment model (Francis & McKenzie 2015) was constructed. The CASAL and AB models comprised three stock areas comparable to East Northland (ENLD), Hauraki Gulf (HAGU), and Bay of Plenty (BPLE), but unlike the actual 2015 SNA 1 assessment model, did not incorporate movement between stock areas. Four harvest methods specified in the models were: Bottom Long Line (BL); Bottom Trawl (BT), Danish Seine (DS) and recreational line (REC). The models covered the period 1900 to 2015 with similar method-area catch histories to those used in the 2013 SNA 1 assessment (Francis & McKenzie 2015). The model stock area B0 values in tonnes were: ENLD 200 000; HAGU 400 000; BPLE: 200 000. The model's main parameter values are given in Table 1. Gear selectivities were length-based and the same in all areas as was growth (Table 1). Growth and selectivity remained constant across the whole catch history period.

Table 1: CASAL and AB model fixed parameters.

Parameter name	Parameter	Input values
Natural mortality	m	0.075 y^{-1}
Stock-recruit steepness (Beverton & Holt)	h	0.85
Proportion mature		0 for ages 1–4, 0.5 for age 5, 1 for ages > 5
Length-weight [mean weight (tonnes) = a (length (cm))^b]		$a = 4.467 \times 10^{-8}$, $b = 2.793$
von Bertalanffy growth parameters		
	L_∞	70.0 cm
	k	0.1 y^{-1}
or as VB growth as equivalent linear increment parameters	*g20*	4.76 cm
	g50	1.90 cm
Assumed growth parameter coefficient of variation	*cv*	*0.2*
Selectivity Long Line {Length based double normal}	$a, \sigma L, \sigma R$	30.47, 2.76, 1000
Selectivity Bottom Trawl {Length based double normal}	$a, \sigma L, \sigma R$	29.39, 2.35, 29.15
Selectivity Danish seine {Length based double normal}	$a, \sigma L, \sigma R$	31.63, 3.13, 20.54
Selectivity Rec {Length based double normal}	$a, \sigma L, \sigma R$	30.11, 1.97, 15.27

Observational data generated by the AB model as input to the CASAL estimator model were:

- Three longline relative abundance indices (one for each area) covering the entire history of the model (1900–2015).
- Proportional catch at-length/age observations for all methods in all areas over the entire history of the model (1900–2015).

4.2 Three-stock length-based CASAL model with identical parameterisation and no parameter estimation

Identically configured and structured CASAL and AB model length-based models produced almost identical spawning stock biomass trajectories (Figure 1). This indicated that the two modelling programs have fundamentally equivalent computational dynamics.

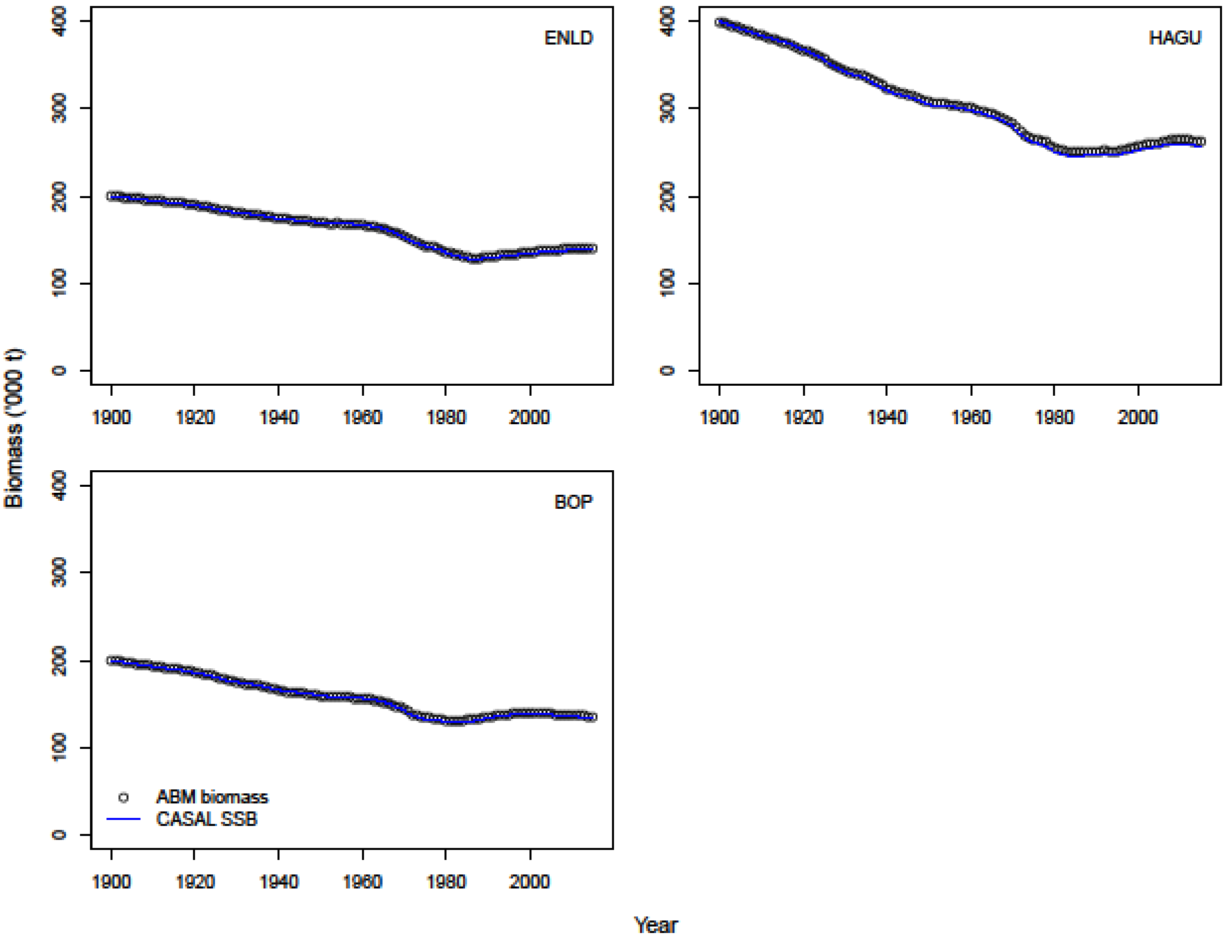


Figure 1: Spawning Stock Biomass trajectories from identically configured and parameterised CASAL length-based and AB models for three stock regions.

4.3 Three-stock length-based CASAL model identical parameterisation, parameter estimation

A length-based CASAL model was fitted to AB model generated CPUE and length-frequency observational data. Parameters estimated in the CASAL model were: Virgin mean Biomass (B0; 3 parameters); gear length-based selectivity (double-normal over four methods; 12 parameters); growth (linear increment, 2 parameters (*g20, g50*)). Parameters not estimated were fixed at the “true” AB model parameter values. Fitting was achieved by placing very tight CVs on the CPUE indices then implementing the Francis reweighting process on the compositional likelihoods (Francis 2011).

CASAL parameter estimates were very close to the “true” AB model parameters (Table 2). The CASAL predicted stock area SSB trajectories were very closely matched to the “true” AB model stock-area SSBs (Figure 2). As CASAL achieved excellent fits to the CPUE and length compositional data (Appendix 6), we believe the slight differences in the CASAL growth parameter estimates (Table 2) are due to the need to balance the CASAL “Minsigma” growth asymptote parameter (refer Bull et al. 2012) so that the CASAL growth dynamic was probably dynamically equivalent to the AB model.

Table 2: True AB model and comparable CASAL parameter estimates after fitting to AB model generated CPUE and length-frequency compositional data. CV on CPUE indices was 0.05; multinomial unscaled length frequencies down-weighted by 90% using Francis (2011) reweighting method.

Parameter name	**Parameter**	**AB model true parameter values**	**CASAL estimates**
Virgin spawning stock biomass	*B0 east Northland*	*200 000 tonnes*	*204 296*
	B0 Hauraki Gulf	*400 000 tonnes*	*405 637*
	B0 Bay of Plenty	*200 000 tonnes*	*202 685*
Linear growth(vB) increment parameters (see Francis 1988)	*vB K (g20)*	*0.1 (g20=4.76 cm)*	*0.12 (g20=5.44)*
	vB Linf (g50)	*70 (g50=1.90 cm)*	*67 (g50=1.95)*
	*Minsigma**	N/A	0.924
Selectivity Long Line {Length based double normal}	*a, σL, σR*	30.47, 2.76, 1000	30.43, 2.55, 977.29
Selectivity Bottom Trawl {Length based double normal}	*a, σL, σR*	29.39, 2.35, 29.15	29.52, 2.19, 27.76
Selectivity Danish seine {Length based double normal}	*a, σL, σR*	31.63, 3.13, 20.54	31.81, 3.01, 19.89
Selectivity Rec {Length based double normal}	*a, σL, σR*	30.11, 1.97, 15.27	30.44, 1.91, 14.83

*** Additional CASAL parameter to control asymptote to prevent negative growth increments above *Linf* (Bull et al. 2012)**

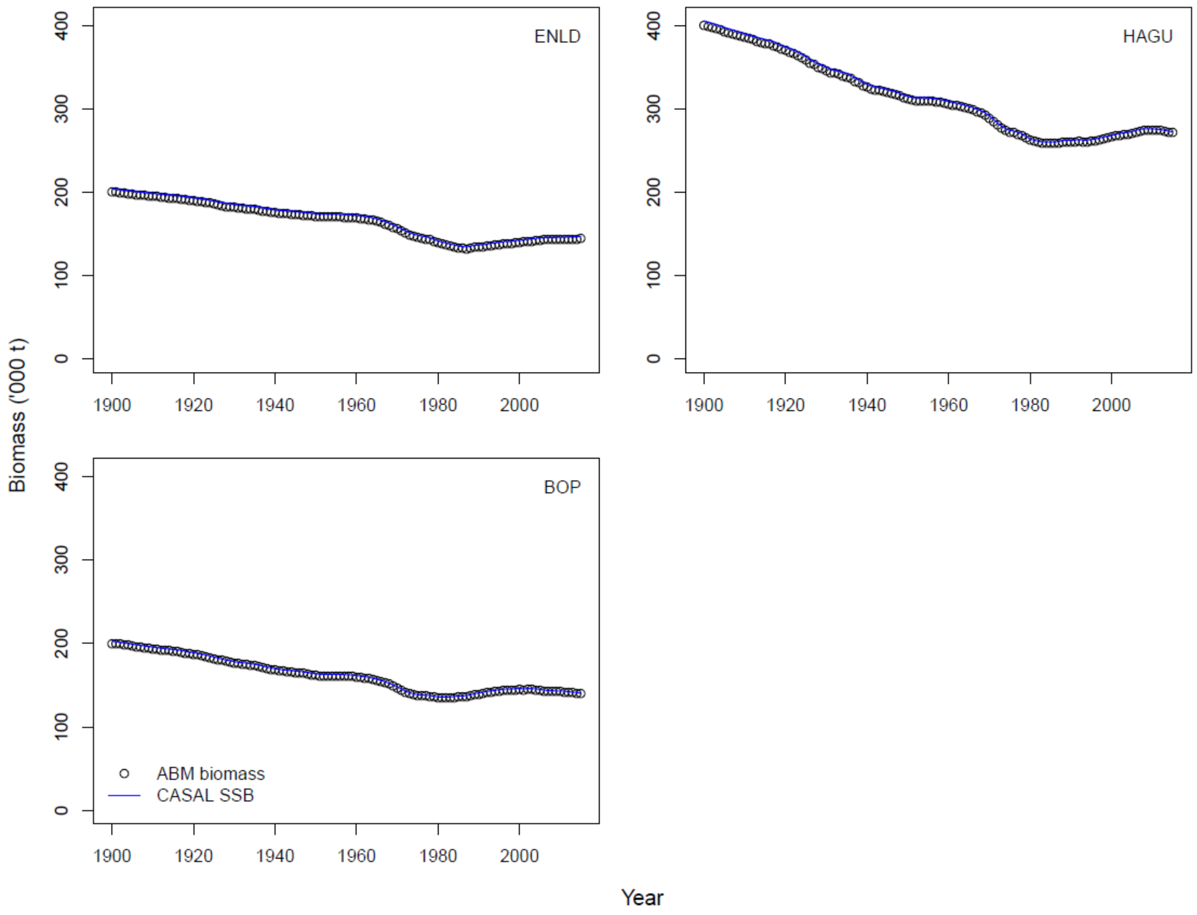


Figure 2: CASAL estimated spawning Stock Biomass trajectories compared to "true" AB model SSBs for three stock regions.

4.4 Three-stock age-based CASAL model fitted to AB model generated observational data with annual recruitment variation

An age-based CASAL model was fitted to AB model generated CPUE and age-frequency observational data. Parameters estimated in the CASAL model were: Virgin mean Biomass (B0; 3 parameters); gear age-based selectivity (double-normal over four methods; 12 parameters); growth (linear increment; 2 parameters (k, l_{inf})); 37 "actual" (unadjusted for steepness) year-class deviates representing the years 1941–1977. Parameters not estimated were fixed at the "true" AB model parameter values.

Again, CASAL parameter estimates were very close to the "true" AB model parameters (Table 3). CASAL stock "actual" (unscaled) year-class strength estimates closely matched the frequency and magnitude of the true AB model stocks (Figure 3).

The CASAL predicted stock area SSB trajectories were a very close match to the "true" AB model stock-area SSBs (Figure 4).

Table 3: True AB model and comparable CASAL parameter estimates after fitting to AB model generated CPUE and age-frequency compositional data. CV on CPUE indices was 0.01; multinomial unscaled length frequencies were down-weighted using the Francis (2011) reweighting method.

Parameter name	**Parameter**	**AB model true parameter values**	**CASAL estimates**
Virgin spawning stock biomass	*B0 east Northland*	*200 000 tonnes*	*196 207*
	B0 Hauraki Gulf	*400 000 tonnes*	*402 661*
	B0 Bay of Plenty	*200 000 tonnes*	*200 357*
Linear growth(vB) increment parameters (see Francis 1988)	*vB K (g20)*	*0.1 (g20=4.76 cm)*	*0.106*
	vB Linf (g50)	*70 (g50=1.90 cm)*	*70.0*
Selectivity Long Line {Length based double normal}	*a, σL, σR*	30.47, 2.76, 1000	4.8(age), 0.77, 911
Selectivity Bottom Trawl {Length based double normal}	*a, σL, σR*	29.39, 2.35, 29.15	4.5(age), 0.65, 17.45
Selectivity Danish seine {Length based double normal}	*a, σL, σR*	31.63, 3.13, 20.54	4.7(age), 0.64, 12.10
Selectivity Rec {Length based double normal}	*a, σL, σR*	30.11, 1.97, 15.27	4.6(age), 0.57, 7.42

The CASAL predicted gear selectivity curves (Figure 5) are consistent in shape to the true length-based curves in the AB model (Appendix 6).

CASAL was able to achieve very close fits to the AB model CPUE indices (Appendix 6). CASAL predicted patterns in mean age by year and region, although closely matching the trends in the AB generated age data, consistently overestimated the mean size for some gear region combinations (Appendix 6).

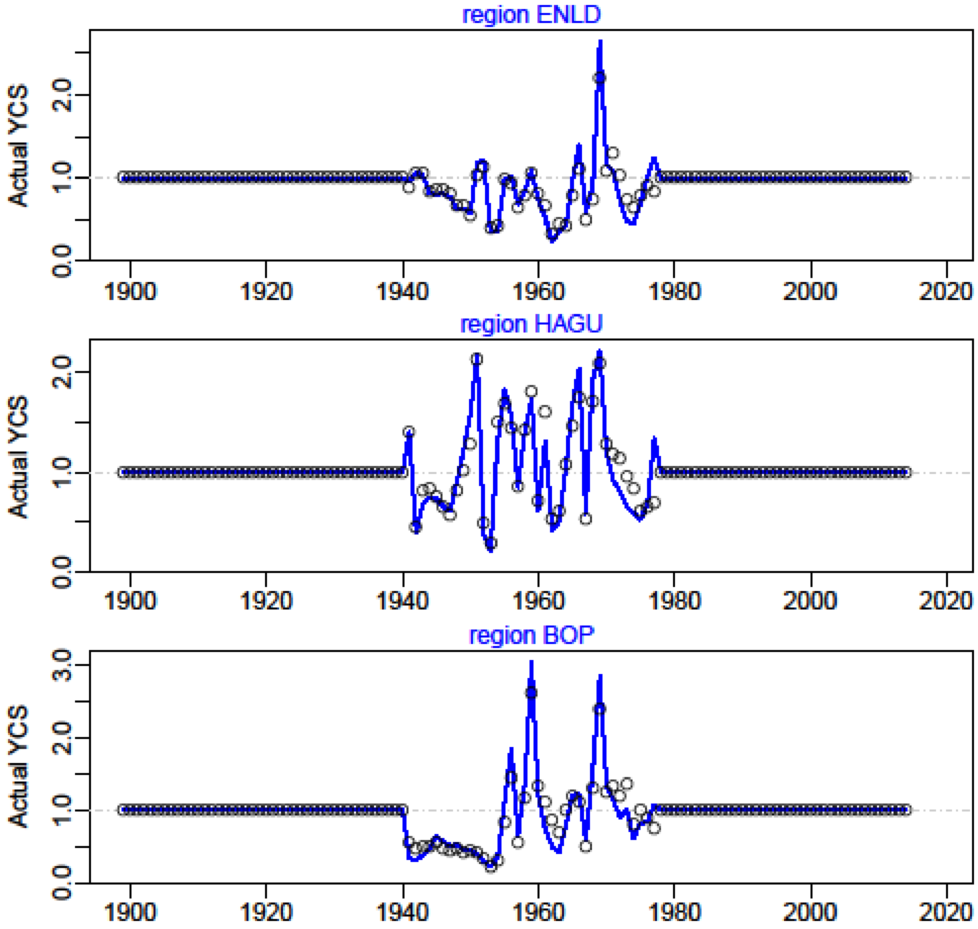


Figure 3: CASAL predicted stock (region) actual year-class strength parameters (blue line) compared to AB model true parameter values (circles) for three stock regions.

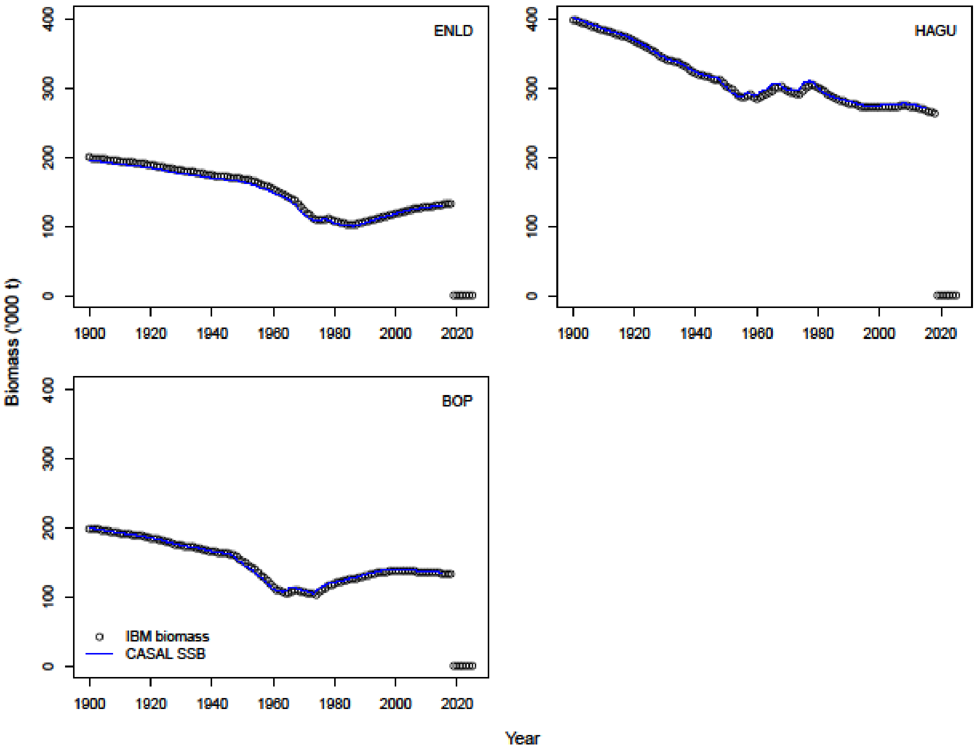


Figure 4: Spawning Stock Biomass trajectories from two identically configured and parameterised CASAL age-based and AB model models for three stock regions.

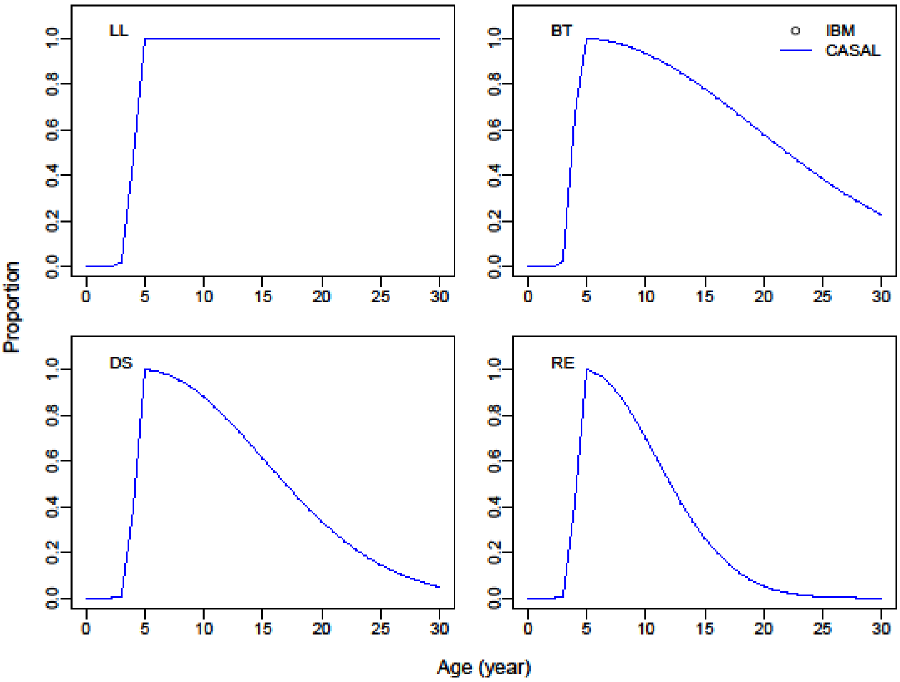


Figure 5: Shape of CASAL estimated age-based selectivity curves for the four model methods as derived from double-normal parameters given in Table 3.

4.5 Petersen tag estimation model predictions of recruited stock size (numbers) growth and movement from AB model generated single tagging event observational data.

The movement and tagging functionalities of the AB model were tested using a single-release multiple-recovery event Petersen tagging (PT) estimator written in R. Given AB model tag release and recovery observations, the PT estimator should be capable of generating accurate population estimates at the time of tagging but to do this it would first "correct" the tag recovery observations for movement between stock areas, and for growth over recovery periods greater than one AB model year.

To correct tag recovery data for growth over recovery time-steps greater than one year, the PT estimator first had to estimate growth. It did this using the tag growth increment data supplied by the AB model. The PT estimator then used the growth estimate to derive a growth transition matrix. This matrix was then used to "back-project" the observed AB model scanned length frequencies to the time-of-tag-release (note: the PT estimator did not need to also back project the recovered tagged agent lengths, as it already had this information from the AB model).

The PT estimator was also able to derive proportional movement estimates algebraically from the AB tag movement data. The PT estimator then used the resultant annual movement proportional matrix to account for subsequent agent movement out to the region they were originally tagged.

The AB model was configured for annual Markovian (diffusive) movement between three stock regions (ENLD; HAGU; BPLE), movement thus parameterised by a 9 cell movement matrix (3 × 3). To get around the added complication of the PT estimator having to estimate a linear growth increment asymptote parameter (analogous to CASAL's "minsigma" growth parameter (Bull et al. 2012)), both the AB model and the PT estimator were instead configured for exponential growth.

The AB model data represented a single tagging event in the year 2000 with tag recoveries occurring over 5 model annual time-steps (2000, 2001, 2002, 2003, and 2004). Note: AB model tag recoveries in the first time-step (2000) were not subject to movement or growth or natural mortality.

The AB tag model population parameterisations and catch history were as described in the previous simulations (Section 4.1.1) with the additional implementation of Markovian movement between the three stock areas (Table 4) and exponential growth (lambda = 0.046, kappa = 9.59; Appendix 2).

Table 4: True AB model annual stock-area proportional Markovian movement parameters

			From
To	ENLD	HAGU	BPLE
ENLD	0.8	0.1	0.1
HAGU	0.1	0.7	0.3
BPLE	0.1	0.2	0.6
Σ	1.0	1.0	1.0

On the basis of the data provided by the AB model, the tagging estimator could provide estimates of the number of **unscaled** agents in the population being based on **unscaled** agent release recovery AB model numbers. To increase the "interpretative power" of the analysis, the number of model agents was set to the computational maximum of 100 million. Recovery observations were pursuant to tagging 100 000 agents in each of the three model areas. All tags were released by longline method pursuant to the method's selectivity characteristics (Table 3; FigureAppendix 8). No agents less than 25 cm were tagged. To avoid added computational complication, tags loss was set at zero as was tag-related release mortality and trap avoidance. Tag recoveries were generated from the commercial fishery harvest pursuant to setting both the scanning catch percentage and tag detection rate at 100%.

The PT estimator regional population estimates were very close to the AB model true values (Figure 6; Table 5). Although the PT estimator growth and movement parameter estimates were not exactly the same (Table 5), these parameter values were sufficiently close to the true AB model values (Appendix 8) to allow the PT estimator to derive population estimates very close to the AB model true values. The PT estimator's excellent performance would also not have been possible if the AB model observational tagging data and movement processes had been erroneous, thus the results can be viewed as validating AB model tagging and movement dynamics.

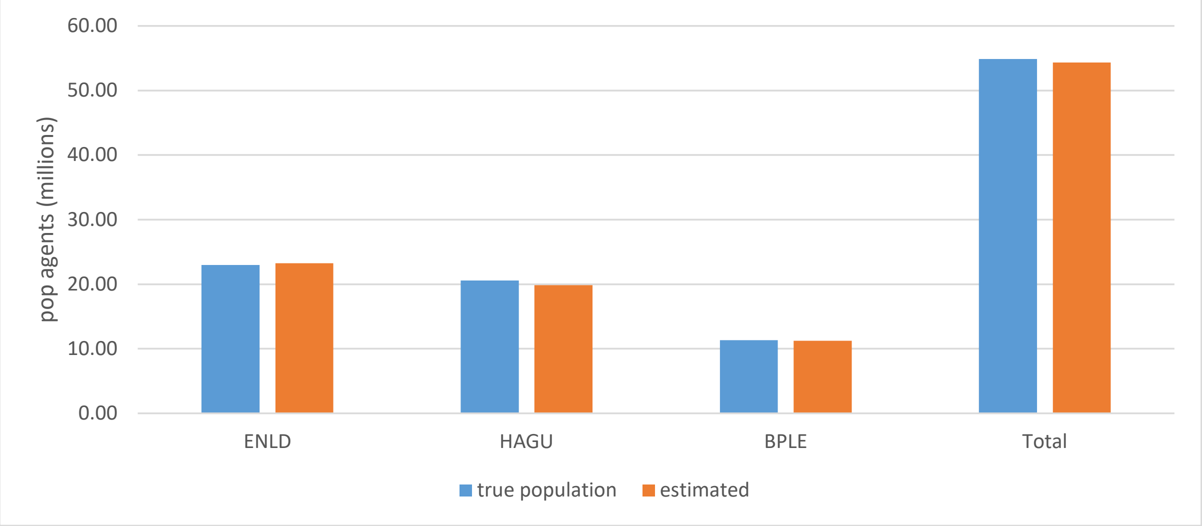


Figure 6: PT estimator derived regional agent population numbers (2000) compared with AB model true values.

Table 5: PT estimator growth and movement parameter estimates and derived 2000 regional population numbers.

	ABM	PT estimator
Growth parameters		
lambda	0.046	0.04762765
kappa	9.59	10.43657
Movement parameters		
EN -> EN	0.8	0.74
EN -> HG	0.1	0.20
EN -> BP	0.1	0.06
HG -> EN	0.1	0.13
HG -> HG	0.7	0.71
HG -> BP	0.2	0.16
BP -> EN	0.1	0.20
BP -> HG	0.3	0.20
BP -> BP	0.6	0.60
Population numbers (2000)		
(no. agents >=25 cm)		
ENLD	23 000 000	23 800 000
HAGU	20 600 000	20 600 000
BPLE	11 300 000	11 600 000
Total	54 900 000	56 000 000

5 FUTURE DEVELOPMENTS

5.1 Code generalisation

The current AB model c++ code is hard-coded for three stock-region areas and a 115 year fishing history. An important and relatively straightforward next step is to make the AB model spatio-temporal partition space fully customisable.

Currently not all stock dynamic parameterisations in the model are alterable spatially and temporally. It is important and necessary that parameters like natural mortality, growth and B0 are capable of changing across AB model time and space (again the code-base requires minimal modification to allow this to happen).

5.2 Implementation of sub-annual time-step partitions

The intention is to allow the user to break the annual cycle into any number of discrete time-steps (e.g. season, month day, etc). This will effectively allow seasonal dynamics to be incorporated into the simulations. Note: Annual time subdivision will be essential for fine-scale movement simulation (Section 5.5 below).

5.3 Ability to make tagged fish true individual agents

As discussed in Section 2.5.2, the current agent scaling process limits the effective number of model tag releases in the simulations. It is proposed to implement a process whereby tagged fish are split out from aggregated agents, effectively becoming true individual agents. The process by which this could be accomplished in the code, although not currently implemented, has been tested and validated in R.

5.4 Code base parallelisation

AB models are well suited to parallelisation. Because the process of stepping through each individual agent is independent of the fate of all other agents this task is able to be run simultaneously on multiple processes. Note: It is relatively straight forward to multi-thread the current AB model code-base.

5.5 Implementation of sub-regional spatial partitioning and movement

The addition of sub-regional-partitions will allow the AB model to simulate local-scale effects such as local-depletion and habitat loss. Movement processes at the sub-regional level will act independently, and in addition to, stock-level movement rules. The sub-regional movement will occur after sub-level movement in the agent fate evaluation sequence (Section 2.2.3), i.e., given that the agent does not migrate to a new stock-region, sub-region movement rules will then be enacted to determine whether the agent remains in its current sub-region or moves to another sub-region, within the current stock-region. Rules governing movement at the sub-regional level will be based on advection and dispersion algorithms (Adam & Sibert 2002) and preference functions (Marsh et al. 2015).

6 CONCLUSIONS

The current AB model is suitable as an operating model for SNA 1 MSE, for generating tagging data to evaluate SNA 1 mark-recapture designs, and for testing SNA 1 stock assessment model performance and bias. With only minor structural changes the current AB model simulator could also be used to undertake model-based MSEs for most of New Zealand commercial fish stocks including other spatially disaggregated stocks such as tarakihi and hoki.

The AB model can also currently be used to investigate how changes to gear selectivity (e.g. introduction of new fishing technologies), discard survival, and legal minimum size might alter population level growth rates through differential mortality on faster or slower growing agents.

We believe adding sub-regional (fine) scale partition structure to the AB model to be an important future development option, as this would give the model the capability to simulate complex spatial management scenarios (e.g. closed areas, spatial gear restrictions and marine protected areas) thus greatly enhancing its management strategy evaluation utility. Included with this would be to add complex movement capability at the sub-regional scale (e.g. through the use of preference functions) and also to allow these to be expressed as alternative characteristics at the agent level (e.g. "resident" and "mobile" character agents). This will give the AB model the power to simulate the stock level effects of changing spatial fishing pressures (e.g. small versus large marine reserves) and to see how these spatial management measures favour different movement character agents in the population and thus stake holder access as a result of local depletion.

7 ACKNOWLEDGEMENTS

This work was contracted and funded by the Ministry for Primary Industries under Objectives 1, 2 & 3 of the research project SNA20107. The authors would like to thank Jingjing Zang and Marc Griffiths for manuscript review and Marianne Vignaux for the final editorial comments.

8 REFERENCES

Adam, M.S.; Sibert, J.R. (2002). Population dynamics and movements of skipjack tuna (*Katsuwonus pelamis*) in the Maldivian fishery: Analysis of tagging data from an advection-diffusion-reaction model. *Aquatic Living Resources 15*: 13–23

Beverton, R.J.H.; Holt, S.J. (1957). On the dynamics of exploited fish populations. *Fisheries Investigations Series 2(19).* 533 p.

Bull, B.; Francis, R.I.C.C.; Dunn, A.; McKenzie, A.; Gilbert, D.J.; Smith, M.H.; Bian, R.; Fu, D. (2012). CASAL (C++ algorithmic stock assessment laboratory): CASAL User Manual v2.30-2012/03/21. *NIWA Technical Report 135.* 280 p.

Francis, R.I.C.C. (1988). Are growth parameters estimated from tagging and age-length data comparable? *Canadian Journal of Fisheries and Aquatic Sciences* 45, 936 – 942.

Francis, R.I.C.C. (2011). Data weighting in statistical fisheries stock assessment models. *Canadian Journal of Fisheries and Aquatic Sciences 68:*1124–1138.

Francis, R.I.C.C.; McKenzie, J.R. (2015). Assessment of the SNA 1 stocks in 2013. *New Zealand Fisheries Assessment Report 2015/76.* 82 p.

Grimm, V.; Railsback, S.F. (2005). Individual-based modeling and ecology. Princeton University Press, Princeton, NJ. 428 p.

McKenzie, J.R.; Gilbert, D; Bian, R. (2011). A simulation analysis to derive cost-optimised mark-recapture designs for estimating the biomass of east coast North Island snapper (*Pagrus auratus*) sub-stocks (SNA 1) *New Zealand Fisheries Assessment Report 2011/39.* 170 p.

Marsh, C.; Sibanda, N.; Dunn, M.; Dunn, A. (2015). A Copula-based Habitat Preference Index in Fish Spatial Population Modelling. *Procedia Environmental Sciences, 27, Spatial Statistics conference 2015, 2–5.*

Matsumoto, M.; Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator" *ACM Transactions on Modeling and Computer Simulation. 8 (1): 3–30.*

Ministry for Primary Industries (2017). Fisheries Assessment Plenary, May 2017: stock assessments and stock status. Compiled by the Fisheries Science Group, Ministry for Primary Industries, Wellington, New Zealand. 1475p.

Ricker, W.E. (1954). Stock and Recruitment *Journal of the Fisheries Research Board of Canada, 11(5): 559–623.*

Seber, G. (1982). The estimation of animal abundance and related parameters. Charles Griffen & Company, London. Steele, C.A.;

Smouse, P.E.; Focardi, S.; Moorcroft, P.R.; Kie, J.G.; Forester, J.D.; Morales, J.M. (2010). Stochastic modelling of animal movement. *Philosophical Transactions of the Royal Society B: Biological Sciences, 365(1550):2201–2211.*

9 APPENDICES

Appendix 1: Basic population movement dynamics

There are two fundamental movement dynamics that can be used to model fish movement; Markovian and Home Fidelity (HF).

Markovian movement

Most animal movement models are Markovian in that the parameters governing individual movement are specific to the area in which the animal currently resides (Smouse et al. 2010). In other words, the animal has no prior knowledge of a home area or of an area it visited in a previous time-step that was better, it only knows the suitability or otherwise of the area in which it currently resides.

Markovian movement is typically modelled with a time dependant proportional shift ($\theta_{i \to j}$). The combination of all possible movements to all areas is represented as a proportional movement matrix (Θ). A fundamental property of Markovian movement is that after multiple applications of the movement matrix (Θ) representing movement of tagged fish over successive time-steps, all tagged fish will eventually achieve an equivalent proportional distribution across all strata independent of which stratum they were originally released from (McKenzie et al. 2011).

Home fidelity movement

Under a home fidelity (HF) movement assumption, movement is an attribute of the individual fish rather than the area in which it currently resides. This invokes the concept that individual fish have a predisposition to regard a particularly area as home, e.g. “A” is my home but on occasion I visit “B”. Adopting a home fidelity assumption implies the existence of cryptic Home populations (H_x). These populations are never observed, but the integration of their combined movements is observed.

HF movement can be modelled as an instantaneous probability $\mathrm{P}[xj]$ being the probability of an area x home fish (H_x) being found in area j. The matrix of home population movement probabilities ($\boldsymbol{\psi}$) across all strata represents the equilibrium distribution of all the home strata populations. The equilibrium distribution of H_x tagged fish is therefore attained in the model in the initial time-step after one application of the $\boldsymbol{\psi}$ movement probability matrix. In other words; after an initial mixing process (deemed by our operating model to be instantaneous) the cumulative spatial distribution of tagged fish will not change over successive time-steps. Unlike Markovian movement the equilibrium distribution of marked animals in the population is dependent upon the initial recruitment (birth) stratum.

Appendix 2: Modelling agent growth and growth variability

Currently two growth models can be specified; linear increment growth (von Bertalanffy) and exponential increment growth.

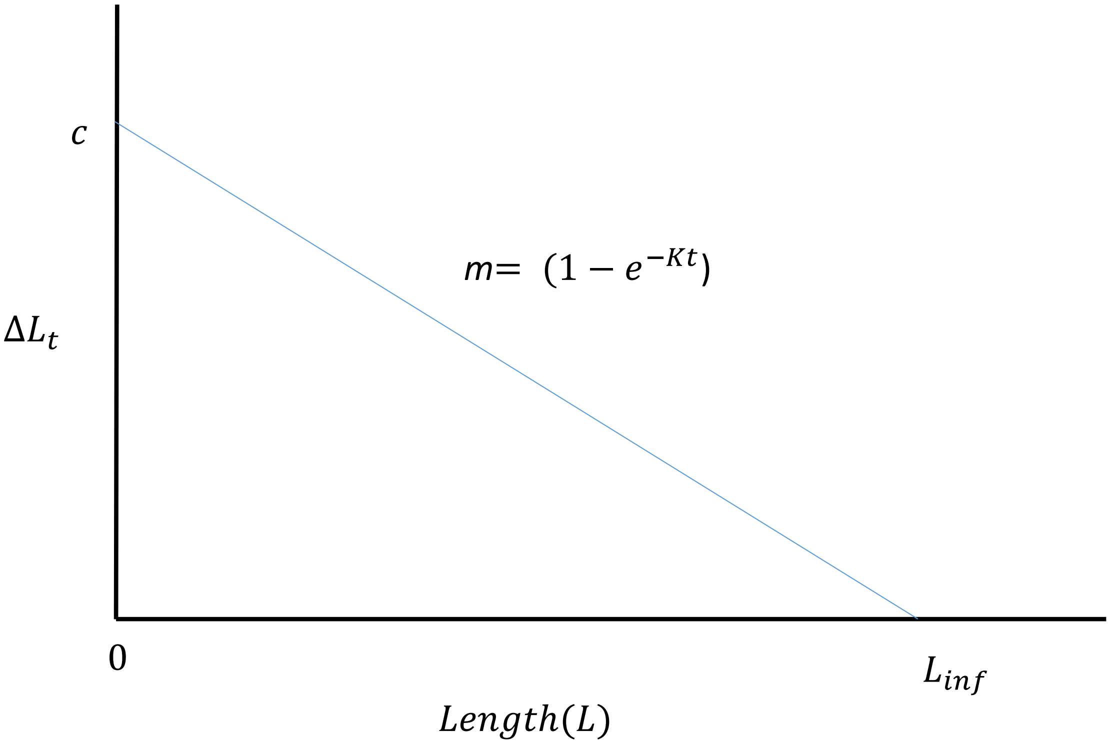
von Bertalanffy (linear increment) growth model

The standard von Bertalanffy growth model which gives the expected length of a fish at age *t* is given by:

$$L_t = L_{inf}(1 - e^{-Kt})$$

Where *K* defines the intrinsic growth rate and L_{inf} is the asymptotic maximum length.

The expected growth increment for a fish of length L with growth path *K* and *Linf* over unit time ***t*** is given by the linear increment formulation of the von Bertalanffy growth model (Francis 1988).



Where:

ΔL = change in length (increment) over unit time t for a fish of given length L_1
m = rate of change of L $(\frac{dL}{dt}) = -(1 - e^{-Kt})$
c = increment for a fish of length = 0 the intercept of $\Delta L = -mL_{inf}$

The change in length (i.e. increment) for a fish of length L_1 over Δt is given by:

$$\begin{aligned}\Delta L_{\Delta t} &= -(1-e^{-K\Delta t})L_1 + c\\ &= (1-e^{-K\Delta t})\left[L_{inf} - L_1\right]\end{aligned}$$

In the AB model, individual growth stochasticity is generated via random draws from lognormal prior distributions for *K* and *Linf* as follows:

K and *Linf* are generated as lognormal random deviates, each fish *y* is assigned its own growth path as specified by K_y and $Linf_y$

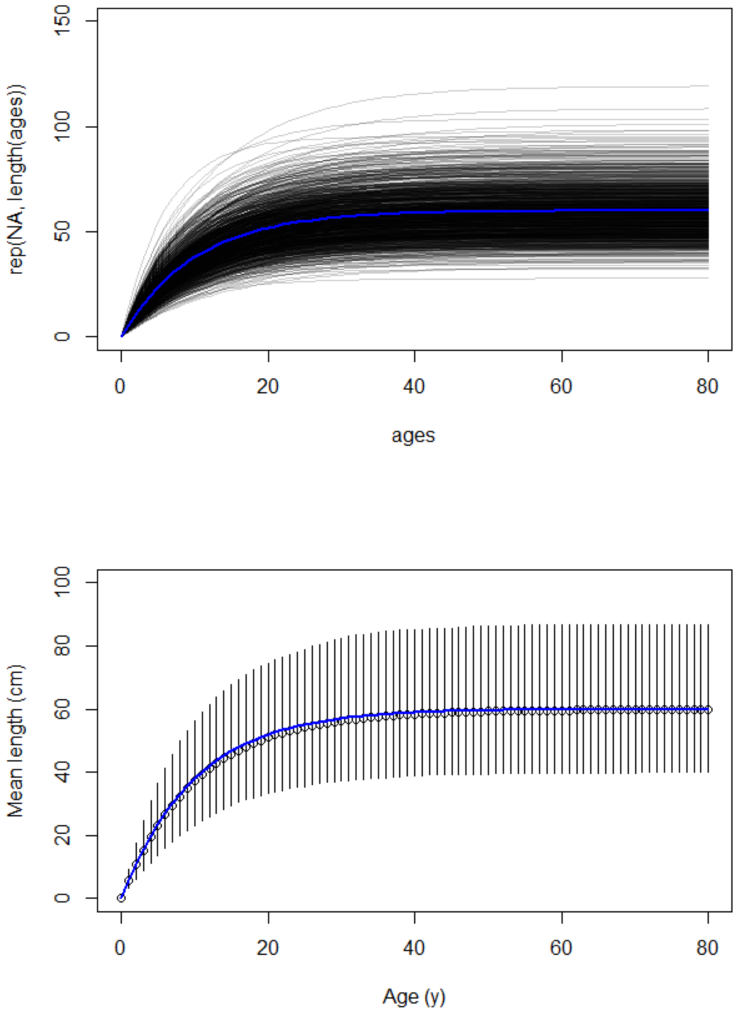
Derivation of *K* ~ (lognormal) assumes K comes from a lognormal distribution with mean = *K* and variance = *v* = (*K**cv)^2.

In log space log(k) comes from a normal distribution with mean = 'u' and standard deviation 's' calculated as follows:

$$u = log\left(K^2/\sqrt{v+K^2}\right)$$
$$s = \sqrt{log\left(\frac{v}{K^2}+1\right)}$$

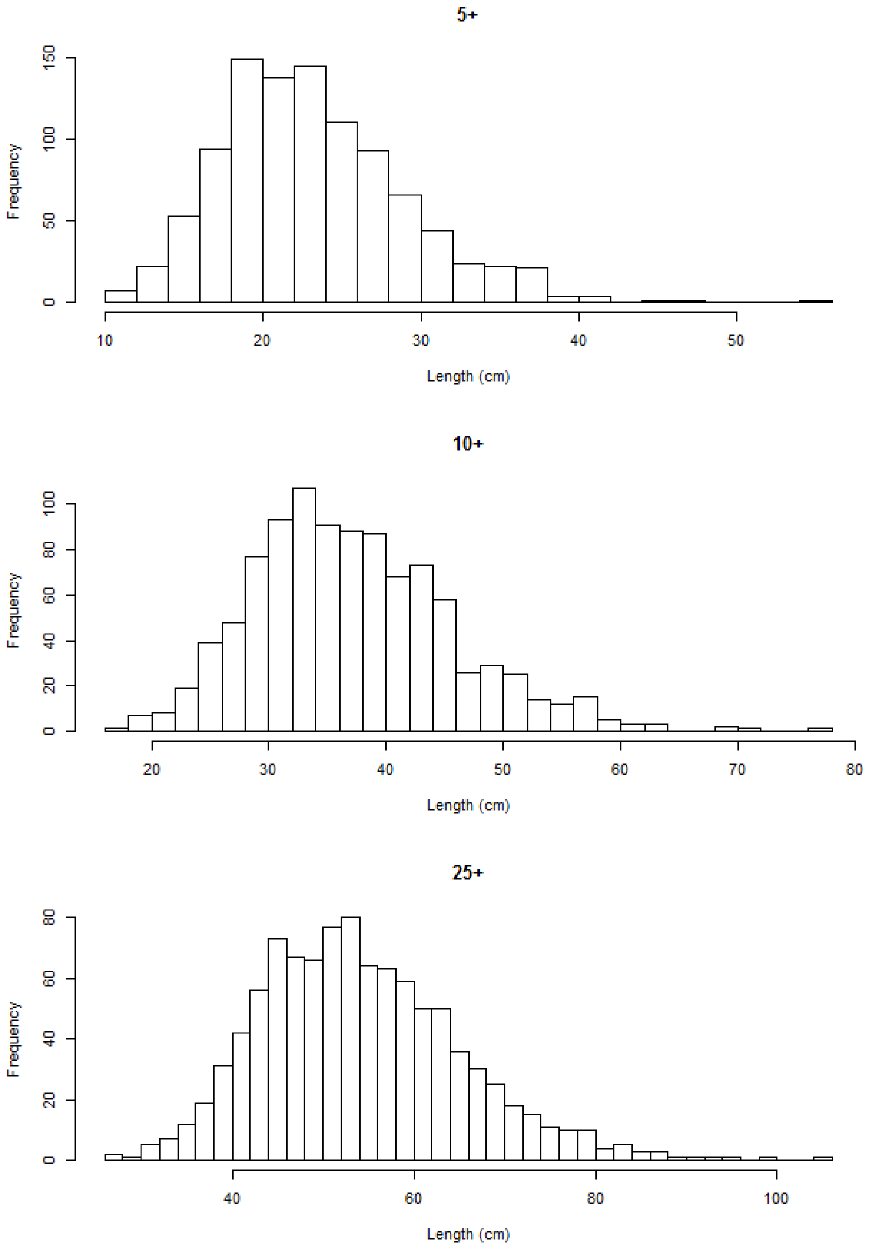
Likewise *log(Linf)~(Normal)*

One thousand individual growth paths were generated randomly from a log-normal distribution of *K* and *Linf* with a CV of 0.2 and respective means 0.1 and 60. The model mean length-at-age as derived from the individual bootstrap growth paths closely matched the actual vB curve (*K* = 0.1 and *Linf* = 60; FigureAppendix 1).



FigureAppendix 1: Individual growth paths from 1000 *K* and *L*$_{inf}$ random draws (top graph). Blue line shows true vB curve for *K* = 0.1 and *L*$_{inf}$ = 60. Dots (bottom graph) are the boot-strap means and vertical lines are bootstrap 95% confidence intervals.

The log normal pattern was evident in the model generated length distributions by age (FigureAppendix 2).



FigureAppendix 2: Model length frequency distributions for age 5, 10, and 25 fish as derived from the 1000 random vB growth paths.

Exponential growth increment model

The parabolic growth model (analogous to the VB model) is specified by two parameters (lambda (λ) and kappa (κ)) given by:

$$L_t = \frac{\log\left[1+\lambda\kappa t e^{(-\lambda L_{t=0})}\right]}{\lambda}$$

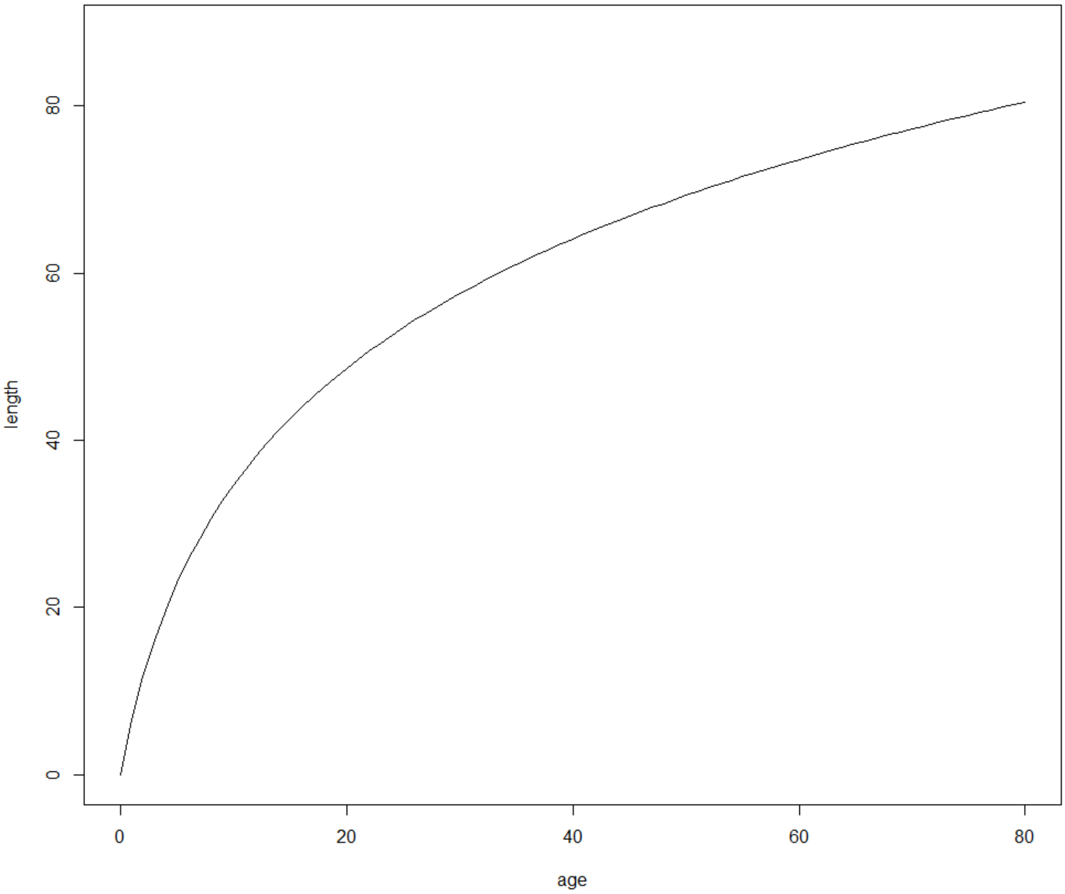
$$\Rightarrow \frac{\log[1+\lambda\kappa t]}{\lambda}$$

For example where

$\lambda = 0.04$

$\kappa = 7.5$

the curve is as follows:



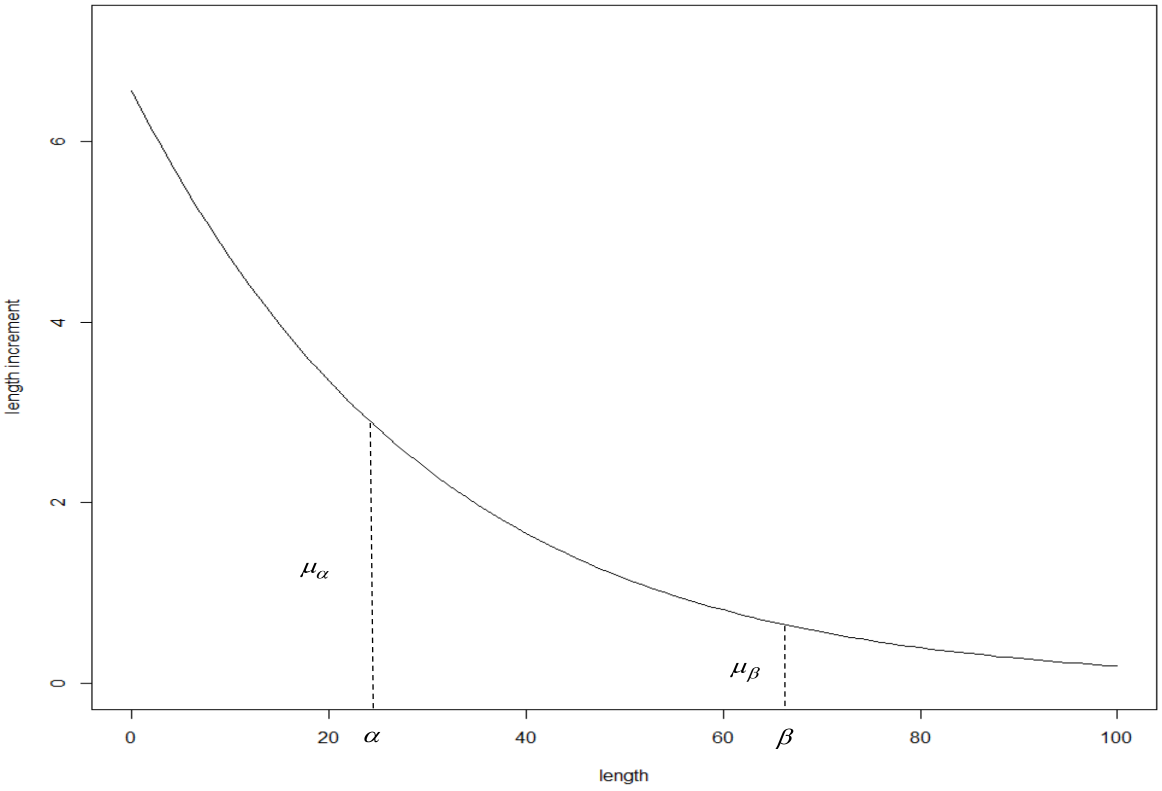
The exponential change in length (i.e. increment) for a fish of length L_1 over Δt is given by:

$$\Delta L_{\Delta t} = \left[\frac{1}{\lambda} \ln \left[1 + \lambda \kappa \Delta t \, \mathrm{e}^{-\lambda L_1} \right] \right]$$

If $\mu_\alpha \,\&\, \mu_\beta$ represent the mean growth increments over Δt for two fish of arbitrary lengths $\alpha \,\&\, \beta$ then:

$$\lambda = \frac{1}{\beta - \alpha} \ln \left[\frac{\mu_\alpha}{\mu_\beta} \right]$$

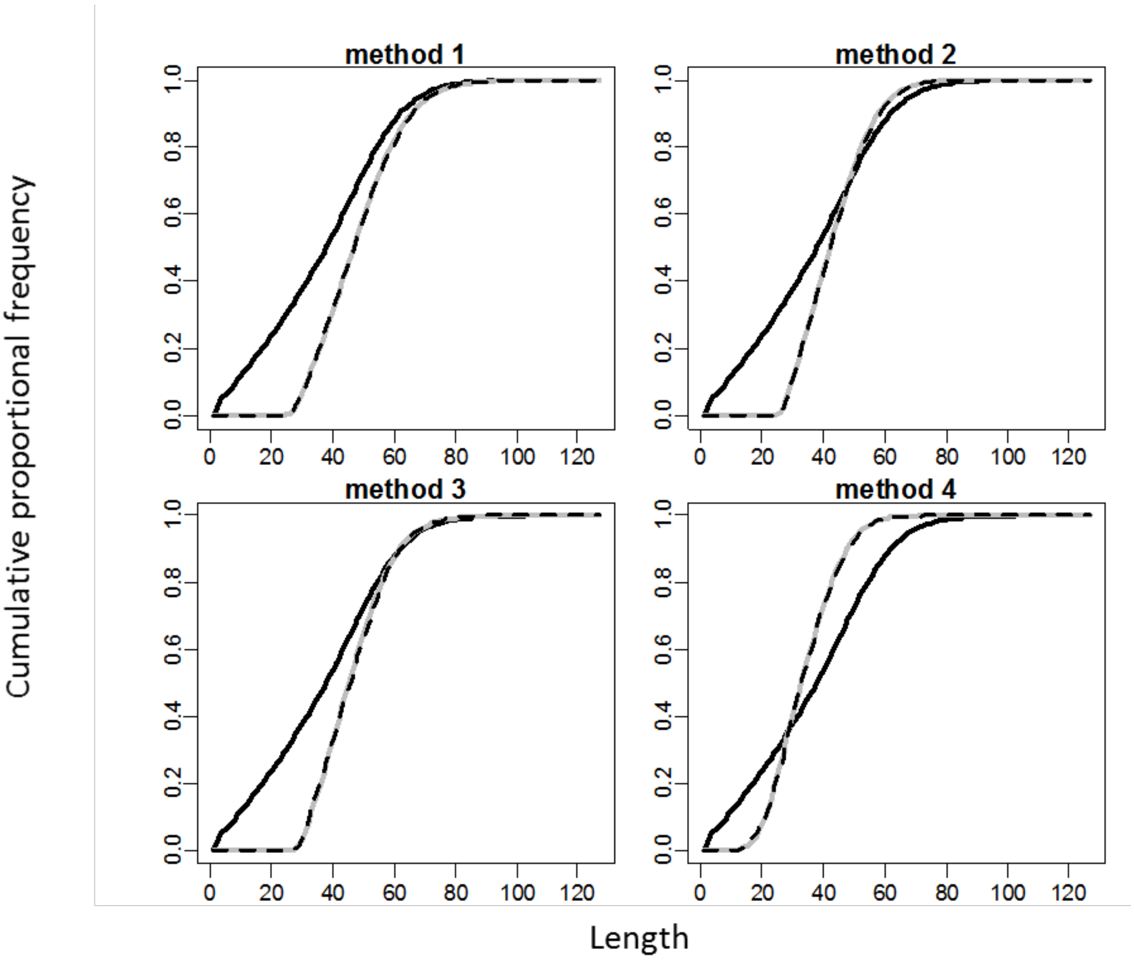
$$\kappa = \mu_\alpha \left(\frac{\mu_\alpha}{\mu_\beta} \right)^{\frac{\alpha}{\beta - \alpha}}$$



As with *K* and *Linf* above, $\lambda_y \,\&\, \kappa_y$ defining the unique parabolic growth curve for each agent *y* are derived as lognormal random deviates of $\lambda \,\&\, \kappa$ as specified by a CV.

Appendix 3: Modelling instantaneous harvest removals by multiple gears

The AB model is able to replicate instantaneous removal length frequency selection by multiple gear types in a given time-step by randomising the gear order in which the probability of capture for each agent is assessed (FigureAppendix 3).



FigureAppendix 3: Instantaneous predicted (grey line) and AB model generated (black segmented line) cumulative catch length frequency distributions taken from a population (black solid line) by methods with differing selectivities.

Appendix 4: AB Model input file details

parameters.json

```
{

"fishes_seed_number":100,000,000 {maximum number of ‘alive’ agents
                                 in the model}
"fishes_seed_z": 0.075, {Determines the equilibrium age structure of
                        the seed population being the equilibrium
                        natural mortality rate}
"fishes_steepness": 0.85, {Beverton and Holt h parameter value}

"fishes_rec_var": 0.6    {sigma-R recruitment variation parameter
                        (refer Bull et al 2012)}
"fishes_males": 0.5, {proportion of agent recruits that are males}

"fishes_m": 0.075, {instantaneous natural mortality rate}


"fishes_a": 4.467e-08,
"fishes_b": 2.793,
                   {length-weight conversion parameters gives
                   unscaled agent weight in tonnes}


"fishes_growth_model": either “l” {linear} or “e” {exponential}
"fishes_k_mean": 0.1,  {vB ‘k’ value if linear; exponent λ value if
                      exponential}
"fishes_k_sd": 0.02, {k or λ standard deviation}
"fishes_linf_mean": 60,{vB ‘Linf’ value if linear; exponent κ value
                       if exponential}
"fishes_linf_sd": 0.05, {Linf or κ standard deviation}

"fishes_growth_variation": either "i" {individual variation} or “t”
                          {temporal variation} or “m” {mixed: both
                          individual and temporal}
"fishes_growth_temporal_cv": 0.1,
"fishes_growth_temporal_sdmin": 1,
"fishes_growth_temporal_incrmin": 0,

"fishes_movement_type":  either “n” {no stock-level movement} or “m”
                        {markovian stock-level movement} or “h”
                        {home-fidelity stock-level level movement}

"harvest_handling_mortality": 0, {discard mortality probability}

"tagging_mortality": 0, {tagging release mortality probability}
"tagging_shedding": 0, {tag loss probability}
"tagging_detection": 1 {tag detection probability}
}
```

fishes_b0.tsv

```
region{stock}  value{tonnes}
0              100000
1              200000
2              100000
```

fishes_movement.tsv

```
region_from    region_to value{agent within-time-step movement
`                             probability}
0              0         0.8
0              1         0.1
0              2         0.1
1              0         0
1              1         0.9
1              2         0.1
2              0         0
2              1         0
2              2         1
```

fishes_rec_strengths.tsv

Year class strengths (multipliers of deterministic recruitment) can be set for each year and region (stock). Use -1 for random recruitment strength (having mean 1 and a CV of fishes_rec_var). Use values of zero or greater to specify a recruitment strength. A recruitment strength value for each year is not required; the default recruitment strength is 1 (i.e. deterministic).

```
year      region    value
1971      0         1.056672289
1972      0         1.270668652
1973      0         1.269415963
1974      0         1.003517688
1975      0         1.014445913
1976      0         1.031370346
1977      0         0.960499135
1978      0         0.80381153
1979      0         0.783037164
1980      0         0.665304626
```

fishes_shyness.tsv

Controls the degree of shyness of a fish to the last fishing method that it was caught by (assuming it was subsequently released because it was undersized or tagged). This is used to mediate its vulnerability to that fishing method in subsequent time-steps. Shyness to a method should be a value between 0 and 1:

- 1 = complete shyness, will never get caught by the method again
- 0 = no shyness, normal vulnerability/selectivity applies

Note that although shyness is of most interest for tagging estimates, it also applies to undersized fish that have been returned to the sea.

```
Method    value
0         0.9
1         0.8
2         0.85
3         0.75
```

harvest_catch_history.tsv

```
Year      Region    Method    value {tonnes}
1900      0         0         233
1901      0         0         311
1902      0         0         245
1903      0         0         344
...
```

monitoring_programme.tsv

Specifies an annual monitoring programme. Each value is a character string with each character specifying if a type of monitoring will be conducted and outputted in output/monitoring:

- C: catch-per-unit-effort index {output to output/monitoring/cpue.tsv}
- L: length sampling of the catch {output to output/monitoring/length_samples.tsv}
- A: age sampling of the catch in {output to output/monitoring/age_samples.tsv}

The default value is an empty string i.e. no monitoring.

```
year        value
1990        CL
1991        CLA
1992        CL
1993        CLA
1994        CL
```

tagging_releases.tsv

Specifies the number of tags to release by year, region and method. Note: tag numbers are unscaled agent numbers and thus represent the actual number of effective unique tags used in the simulation, subsequent up-scaling of this number will not increase estimation precision.

```
year        region       method         value
2018        0            0              10000
2018        1            0              10000
2018        2            0              10000
```

tagging_scanning.tsv

Specifies the proportion of catch to be scanned by year, region and method.

```
year        region       method         value
2018        0            0              1
2018        0            1              1
2018        0            2              0.5
2018        0            3              0.8
```

Appendix 5: Alterable model specifications specified in parameters.hpp

Currently there are still some model specification parameters that can only be changed by editing the "parameters.hpp" c++ file where all the AB model fixed parameters are specified and recompiling. The recompilation requirement will be rectified in the next iteration of the code.

Current hard-coded parameters alterable by changing their values in this file code and recompiling are:

- Gear selectivity (four gear type options: length based and three parameters with double-normal distribution)
- Maturation (single age-based ogive)

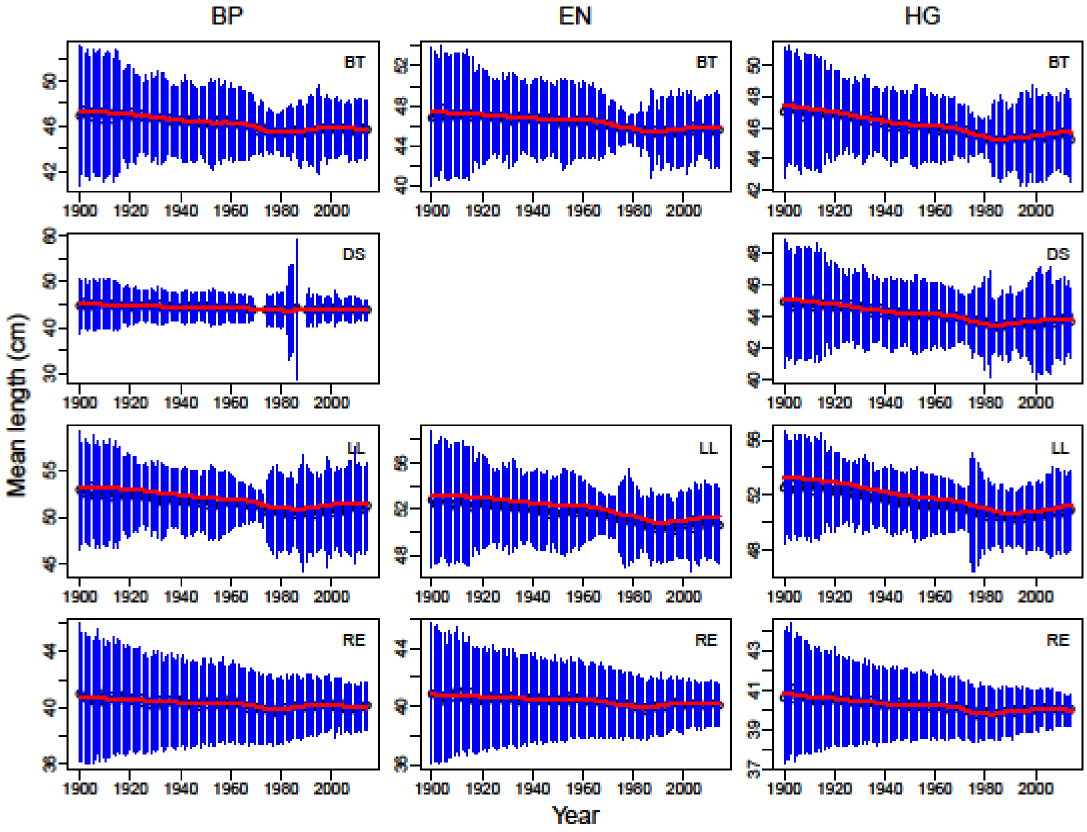
```
#pragma once

#include "requirements.hpp"
#include "random.hpp"
#include "dimensions.hpp"


/**
 * Monitoring components
 *
 * Provides a convienient yet computationally efficient way of specifying
 * annual monitoring programme
 */
class MonitoringComponents : public std::string {
 public:

    bool C, L, A;

    MonitoringComponents(const char* value = ""):
        std::string(value) {
        update();
    }

    void update() {
        C = find('C') != std::string::npos;
        L = find('L') != std::string::npos;
        A = find('A') != std::string::npos;
    }
};

/**
 * Parameters
 *
 * Provides for the input, mapping and output of
 * parameters of the model
 */
class Parameters : public Structure<Parameters> {
 public:

    /**
     * Number of instances of `Fish` to seed the population with
     *
     * Preliminary sensitivity analyses (see `instances_seed_sensitivity` in
`sna1.cpp`)
```

```
     * suggested 100,000 was a good trade-off between run duration and precision at least
     * during development. Should be increased for final runs.
     */
    unsigned int fishes_seed_number = 1e6;

    Uniform fishes_seed_region_dist;

    /**
     * Total mortality of the initial seed population
     *
     * Determines the equilibrium age structure of the seed population.
     */
    double fishes_seed_z = 0.075;

    /**
     * Exponential distribution for ages of the seed population
     */
    Exponential fishes_seed_age_dist;

    /**
     * Pristine spawner biomass (t)
     */
    Array<double, Regions> fishes_b0 = {
        100000,
        200000,
        100000
    };

    /**
     * Stock recruitment
     */
    double fishes_steepness = 0.85;

    /**
     * Recruitment variability
     */
    double fishes_rec_var = 0.6;
    Array<double, Years, Regions> fishes_rec_strengths = 1;

    /**
     * Sex ratio
     */
    double fishes_males = 0.5;

    /**
     * Instantaneous rate of natural mortality
     */
    double fishes_m = 0.075;

    /**
     * Probabiliy of fish dying of natural mortality
     * in a time-step. Derived from `fishes_m` in `initialise()`
     */
    double fishes_m_rate;

    /**
     * Length-weight relation
     */
    double fishes_a = 4.467e-08;
    double fishes_b = 2.793;
```

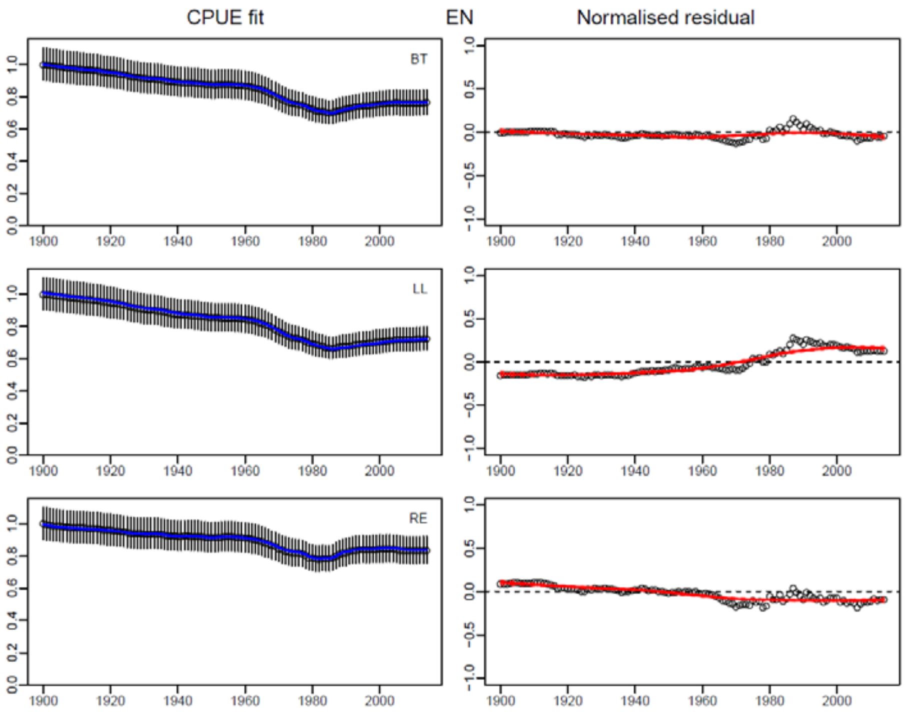
```
/**
 * Growth model
 *
 * l = linear
 * e = exponential
 */
char fishes_growth_model = 'l';

/**
 * Distribution of growth coefficients (von Bertalanffy k)
 * across population of fish
 */
double fishes_k_mean = 0.1;
double fishes_k_sd = 0.02;
Lognormal fishes_k_dist;

/**
 * Distribution of assymptotic length (von Bertalanffy Linf)
 * across population of fish
 */
double fishes_linf_mean = 60;
double fishes_linf_sd = 10;
Lognormal fishes_linf_dist;

/**
 * Growth variation type
 *
 * t = only temporal variation in growth
 * i = only individual variation in growth
 * m = mixed, both individual and temporal variation in growth
 */
char fishes_growth_variation = 'm';

/**
 * Coefficient of variation of temporal variation in growth
 */
double fishes_growth_temporal_cv = 0.3;
double fishes_growth_temporal_sdmin = 1;
double fishes_growth_temporal_incrmin = 0;

/**
 * Maturiation-at-age
 *
 * This is NOT the proportion mature at an age but rather the probability
 * of maturing at a particular age
 */
Array<double, Ages> fishes_maturation;

/**
 * Movement type
 */
char fishes_movement_type = 'm';

/**
 * Movement matrix
 */
Array<double, Regions, RegionTos> fishes_movement = 0;

/**
 * The degree of shyness of a fish to the last fishing method that it
```

```
     * was caught by (assuming it was released because undersized or tagged).
     *
     * 1 = complete shyness, will never get by the method again
     * 0 = no shyness
     */
    Array<double, Methods> fishes_shyness = 0;

    /**
     * Catch history
     */
    Array<double, Years, Regions, Methods> harvest_catch_history = 0;

    /**
     * Minimum legal size limit
     */
    Array<double, Methods> harvest_mls = {
        25, 25, 25, 25
    };

    /**
     * Mortality of fish that are returned to sea
     */
    double harvest_handling_mortality = 0;

    /**
     * Parameters of double normal length based selectivity
     */
    Array<double, Methods> harvest_sel_steep1;
    Array<double, Methods> harvest_sel_mode;
    Array<double, Methods> harvest_sel_steep2;

    /**
     * Monitoring programme by year
     */
    Array<MonitoringComponents, Years> monitoring_programme = "";

    /**
     * The number of target tagging releases by year, region and method
     */
    Array<int, Years, Regions, Methods> tagging_releases = 0;

    /**
     * The proportion of catch scanned by year, region and method
     */
    Array<double, Years, Regions, Methods> tagging_scanning = 0;

    /**
     * Mortality of fish that have been tagged (note, this is independent of
`harvest_handling_mortality`)
     */
    double tagging_mortality = 0;

    /**
     * Probability that a tag will be shed by the fish
     */
    double tagging_shedding = 0;

    /**
     * The probability that a tagged fish is detected when scanned
     */
    double tagging_detection = 1;
    /**
     * Initialise parameters
     */
    void initialise(void) {
        // Defaults not defined above
        fishes_movement = 0;

        for (auto age : ages) {
            double p = 0;
            if (age <= 4) p = 0;
            else if (age == 5) p = 0.5;
            else p = 1;
            fishes_maturation(age) = p;
        }

        harvest_sel_steep1(LL) = 2.76;
        harvest_sel_mode(LL) = 30.47;
        harvest_sel_steep2(LL) = 1000;

        harvest_sel_steep1(BT) = 2.35;
        harvest_sel_mode(BT) = 29.39;
        harvest_sel_steep2(BT) = 29.15;

        harvest_sel_steep1(DS) = 3.13;
        harvest_sel_mode(DS) = 31.63;
        harvest_sel_steep2(DS) = 20.54;

        harvest_sel_steep1(RE) = 1.97;
        harvest_sel_mode(RE) = 30.11;
        harvest_sel_steep2(RE) = 15.27;

        // Parameter values can be overidden by setting them in the following
files:

        #define IFE(FILE, WHAT) if(boost::filesystem::exists(FILE))
WHAT(FILE)

        IFE("input/parameters.json", read);
        IFE("input/fishes_b0.tsv", fishes_b0.read);
        IFE("input/fishes_rec_strengths.tsv", fishes_rec_strengths.read);
        IFE("input/fishes_movement.tsv", fishes_movement.read);
        IFE("input/fishes_shyness.tsv", fishes_shyness.read);
        IFE("input/harvest_mls.tsv", harvest_mls.read);
        IFE("input/harvest_catch_history.tsv", harvest_catch_history.read);
        IFE("input/monitoring_programme.tsv", monitoring_programme.read);
        IFE("input/tagging_releases.tsv", tagging_releases.read);
        IFE("input/tagging_scanning.tsv", tagging_scanning.read);

        #undef IFE

        // Derived values

        fishes_seed_region_dist = Uniform(0,3);
        fishes_seed_age_dist = Exponential(fishes_seed_z);

        fishes_m_rate = 1 - std::exp(-fishes_m);

        fishes_k_dist = Lognormal(fishes_k_mean, fishes_k_sd);
        fishes_linf_dist = Lognormal(fishes_linf_mean, fishes_linf_sd);
        for (auto& item : monitoring_programme) item.update();
    }

    void finalise(void) {
        boost::filesystem::create_directories("output");

        write("output/parameters.json");
        fishes_b0.write("output/fishes_b0.tsv");
        fishes_rec_strengths.write("output/fishes_rec_strengths.tsv");
        fishes_movement.write("output/fishes_movement.tsv");
        fishes_shyness.write("output/fishes_shyness.tsv");
        harvest_mls.write("output/harvest_mls.tsv");
        harvest_catch_history.write("output/harvest_catch_history.tsv");
        monitoring_programme.write("output/monitoring_programme.tsv");
        tagging_releases.write("output/tagging_releases.tsv");
        tagging_scanning.write("output/tagging_scanning.tsv");
    }

    template<class Mirror>
    void reflect(Mirror& mirror){
        mirror
            .data(fishes_seed_number, "fishes_seed_number")
            .data(fishes_seed_z, "fishes_seed_z")

            .data(fishes_steepness, "fishes_steepness")
            .data(fishes_rec_var, "fishes_rec_var")

            .data(fishes_males, "fishes_males")

            .data(fishes_m, "fishes_m")

            .data(fishes_a, "fishes_a")
            .data(fishes_b, "fishes_b")

            .data(fishes_growth_model , "fishes_growth_model")
            .data(fishes_k_mean , "fishes_k_mean")
            .data(fishes_k_sd, "fishes_k_sd")
            .data(fishes_linf_mean, "fishes_linf_mean")
            .data(fishes_linf_sd, "fishes_linf_sd")
            .data(fishes_growth_variation , "fishes_growth_variation")
            .data(fishes_growth_temporal_cv , "fishes_growth_temporal_cv")
            .data(fishes_growth_temporal_sdmin,
"fishes_growth_temporal_sdmin")
            .data(fishes_growth_temporal_incrmin,
"fishes_growth_temporal_incrmin")

            .data(fishes_movement_type, "fishes_movement_type")

            .data(harvest_handling_mortality, "harvest_handling_mortality")

            .data(tagging_mortality, "tagging_mortality")
            .data(tagging_shedding, "tagging_shedding")
            .data(tagging_detection, "tagging_detection")
        ;
    }
 };  // class Parameters
```

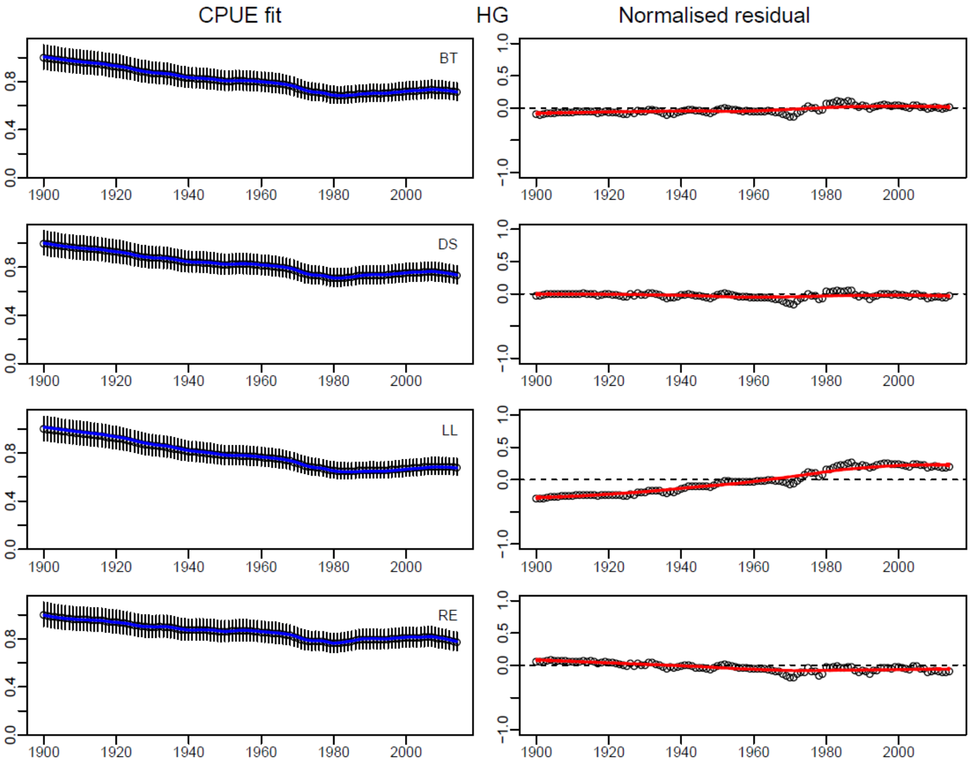
Appendix 6: CASAL length-based model fits to AB model CPUE and length-frequency observational data



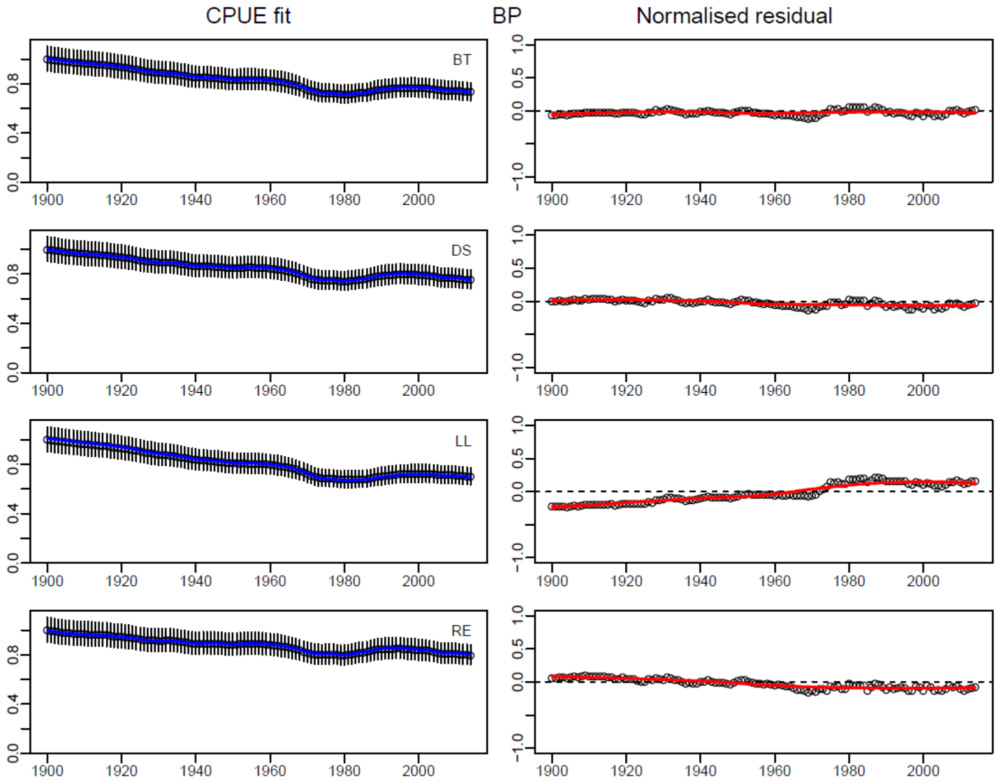
FigureAppendix 4: Mean length with standard error bars of AB model generated length data by area and method. Red line shows CASAL predicted mean lengths.



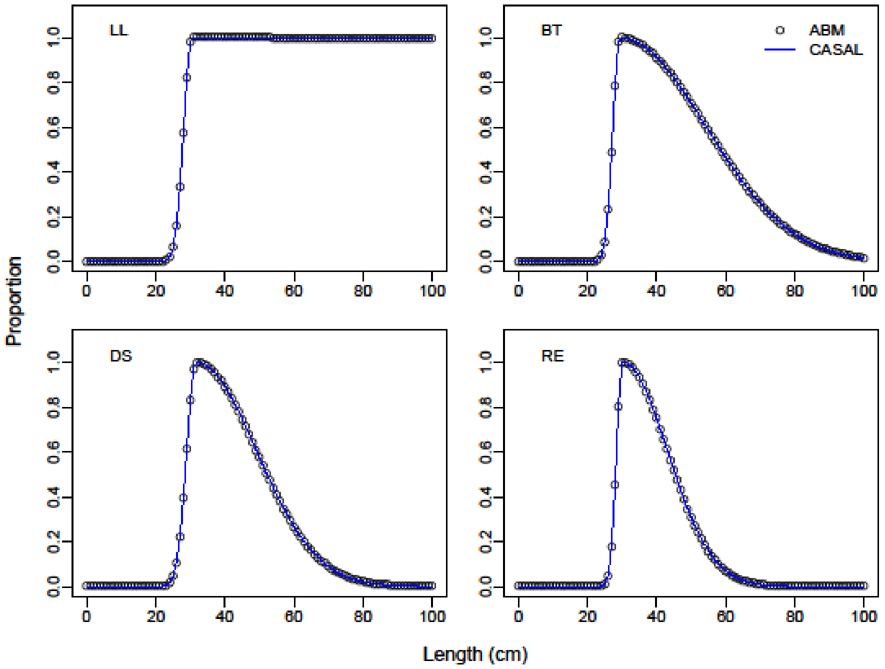
FigureAppendix 5: CASAL fits (blue line) to AB model East Northland area (EN) method specific CPUE indices (error bars reflect assumed CV of 0.05). Model normalised residual plots with fitted spline curves shown in red.



FigureAppendix 6: CASAL fits (blue line) to AB model Hauraki Gulf area (HG) method specific CPUE indices (error bars reflect assumed CV of 0.05). Model normalised residual plots with fitted spline curves shown in red.

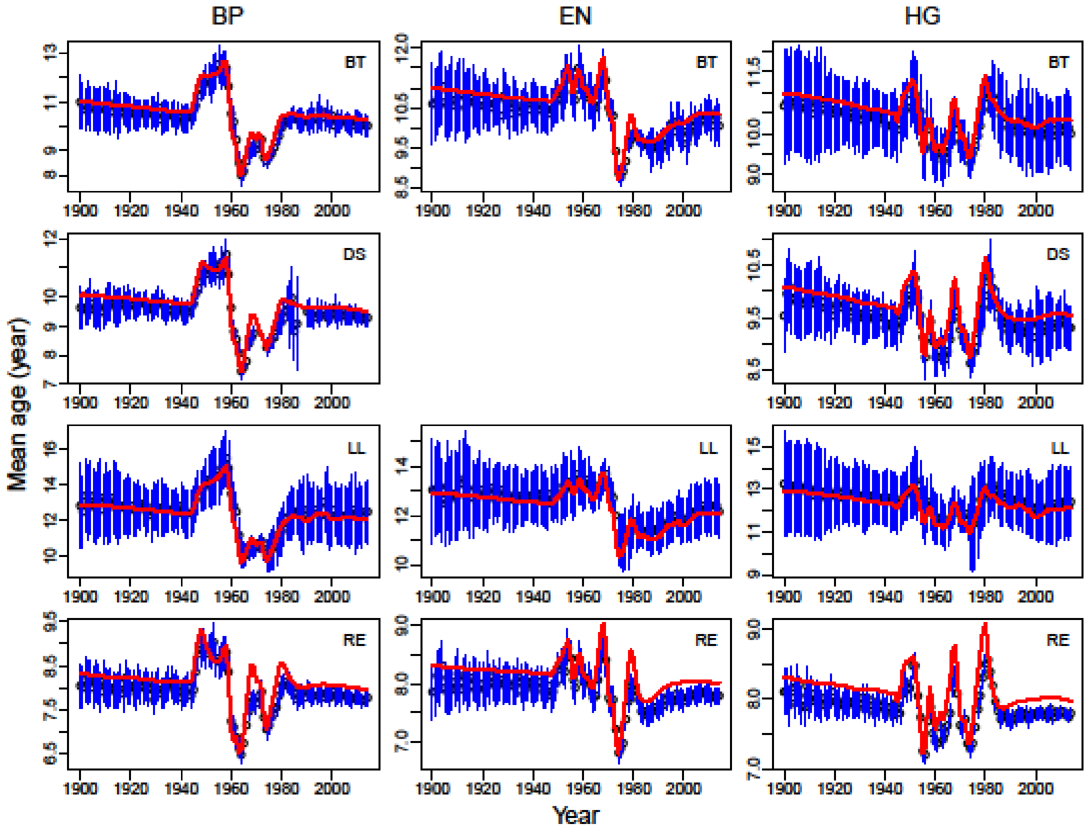


FigureAppendix 7: CASAL fits (blue line) to AB model Bay of Plenty area (BP) method specific CPUE indices (error bars reflect assumed CV of 0.05). Model normalised residual plots with fitted spline curves shown in red.

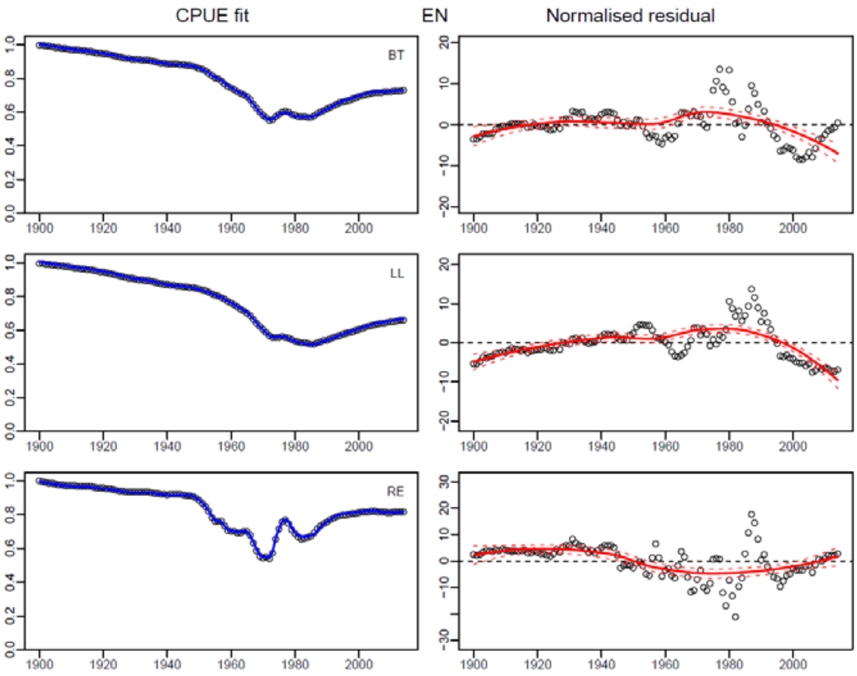


FigureAppendix 8: CASAL estimated selectivity curves (blue line) superimposed on AB model “actual” gear selectivity curves (circles).

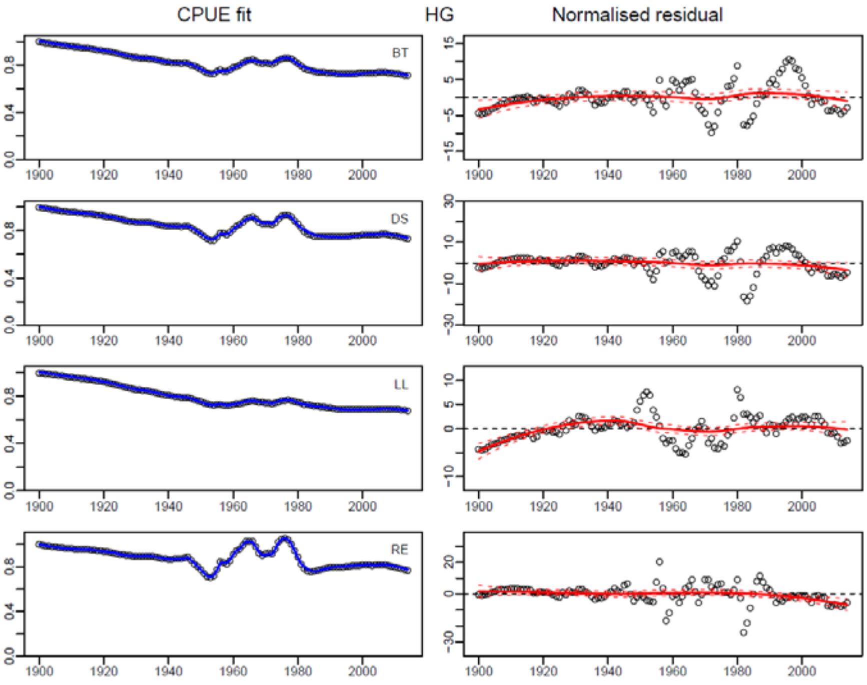
Appendix 7: CASAL Age-based model fits to AB model CPUE and age-frequency observational data



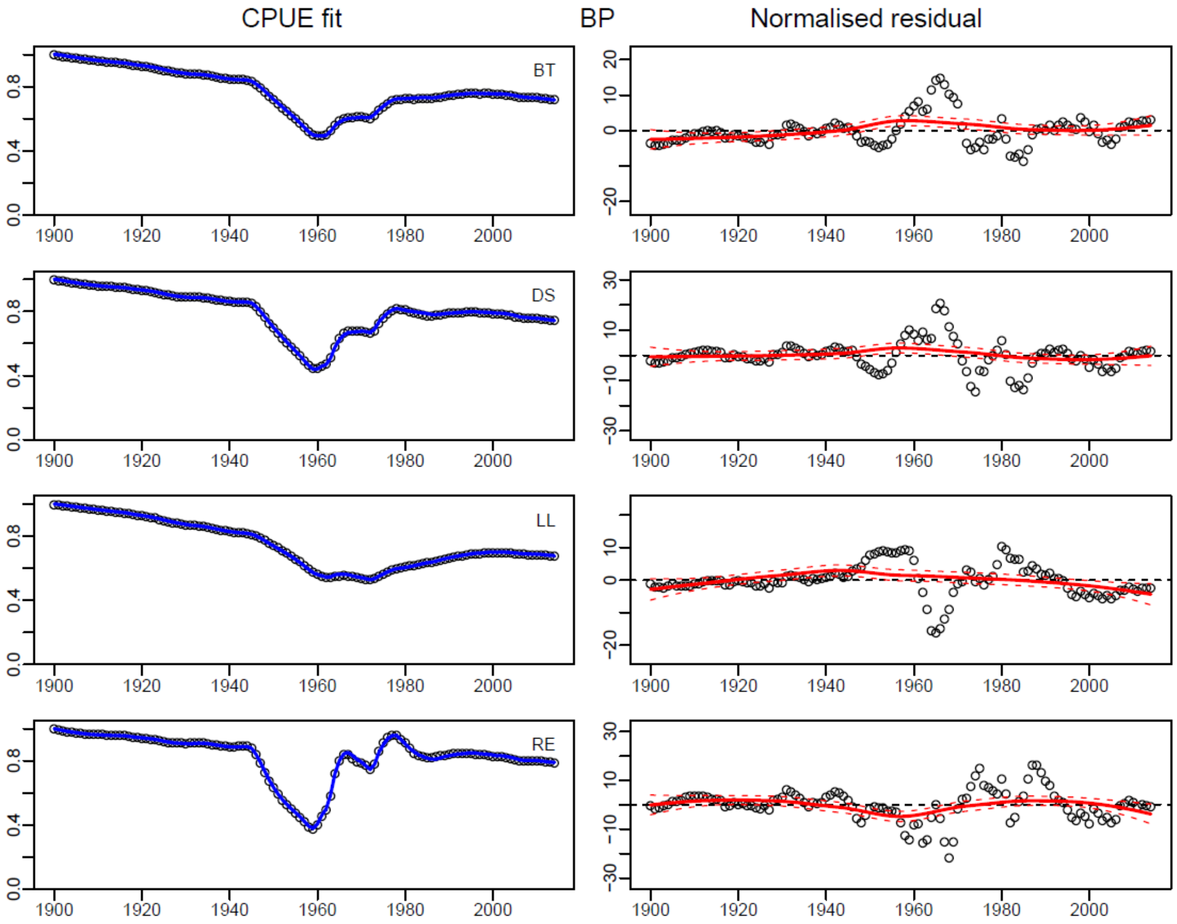
FigureAppendix 9: Mean age with standard error bars of AB model generated length data by area and method. Red line shows CASAL predicted mean ages.



FigureAppendix 10: CASAL fits (blue line) to AB model East Northland area (EN) method specific CPUE indices (error bars reflect assumed CV of 0.01). Model normalised residual plots with fitted spline curves shown in red.

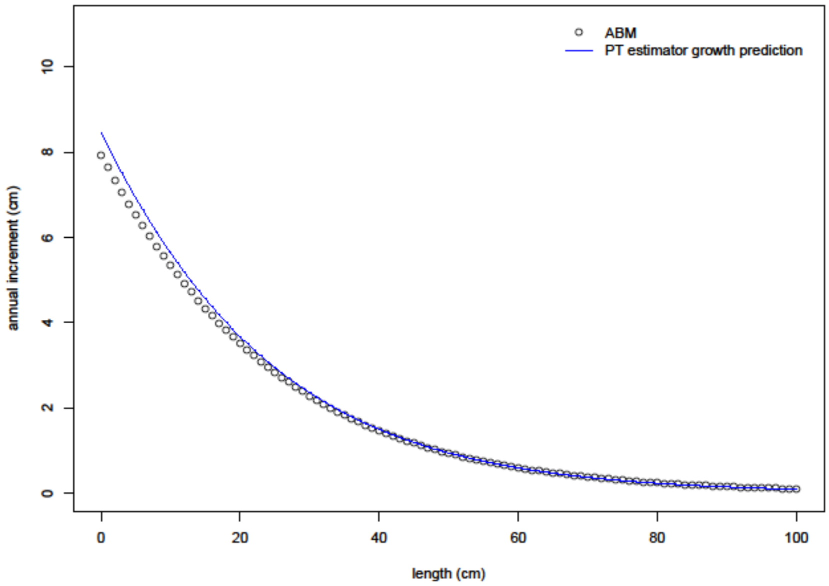


FigureAppendix 11: CASAL fits (blue line) to AB model Hauraki Gulf area (HG) method specific CPUE indices (error bars reflect assumed CV of 0.01). Model normalised residual plots with fitted spline curves shown in red.

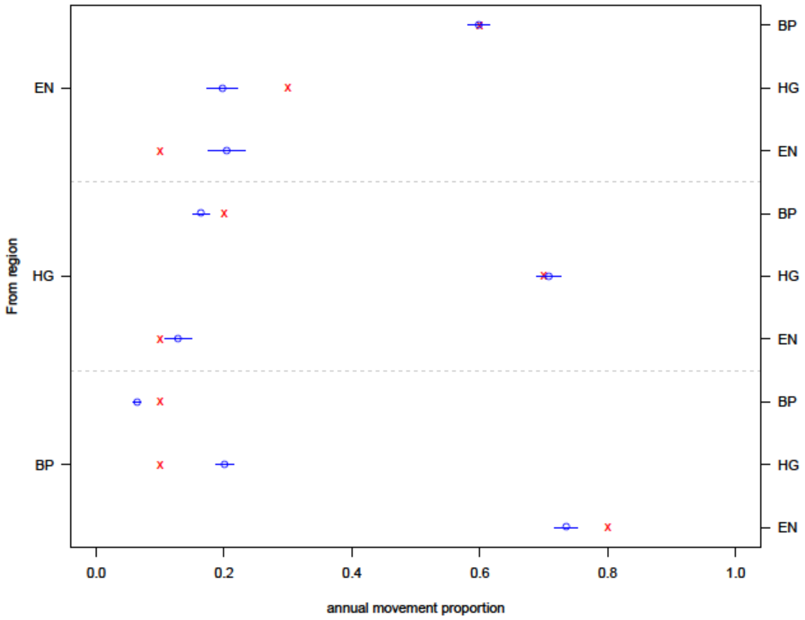


FigureAppendix 12: **CASAL fits (blue line) to AB model Bay of Plenty area (BP) method specific CPUE indices (error bars reflect assumed CV of 0.01). Model normalised residual plots with fitted spline curves shown in red.**

Appendix 8: PT estimator growth and movement prediction performance compared to true AB model parameterisations



FigureAppendix 13: Comparison of AB model exponential annual growth increment curve to that predicted from the PT estimator fits to the AB model tag growth increment data.



FigureAppendix 14: PT estimator proportional region movement estimates (blue dots with standard error) compared to AB model true values (red X).